

Advances in Computing Science

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov, CIC-IPN, Mexico
Gerhard X. Ritter, University of Florida, USA
Jean Serra, Ecole des Mines de Paris, France
Ulises Cortés, UPC, Barcelona, Spain

Associate Editors:

Jesús Angulo, Ecole des Mines de Paris, France
Jihad El-Sana, Ben-Gurion Univ. of the Negev, Israel
Alexander Gelbukh, CIC-IPN, Mexico
Ioannis Kakadiaris, University of Houston, USA
Petros Maragos, Nat. Tech. Univ. of Athens, Greece
Julian Padget, University of Bath, UK
Mateo Valero, UPC, Barcelona, Spain
Rafael Guzmán, Univ. of Guanajuato, Mexico

Editorial Coordination:

Alejandra Ramos Porras
Carlos Vizcaino Sahagún

Research in Computing Science, Año 18, Volumen 148, No. 10, octubre de 2019, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. <https://www.rcs.cic.ipn.mx>. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 30 de agosto de 2019.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politécnico Nacional.

Research in Computing Science, year 18, Volume 148, No. 10, October 2019, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research of the IPN.

Advances in Computing Science

Juan Carlos Chimal Eguía
Elizabeth López Lozada
Federico Furlán Colín (eds.)



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2019

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2019

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

Editorial

Computer Science is pervasive to the scientific endeavor. There is no area or field of study which, nowadays, can thrive without the use of computational resources, models and algorithms to analyze ever growing amounts of data in increasingly complex ways. This trend is clearly reflected in the selection of papers included in this volume of *Research in Computing Science*.

This issue presents both new and improved algorithms and methods as well as applications to other fields of study. Accepted works range from improved algorithms for solving classical computational problems, to the application of known models and techniques on biomedical challenges.

All submitted papers were reviewed by two or more independent members of the editorial committee. This volume contains revised and corrected versions of the 31 accepted papers.

Our deepest gratitude goes to all the parties involved in the creation of this volume: Foremost to the authors of the articles for their dedication to the excellence of the works presented. We are also grateful for the hard labor the members of the editorial board invested in the evaluation and selection of the highest quality papers amongst many others of high value. Our special and profound gratefulness to the Centro de Investigación en Computación of the Instituto Politécnico Nacional (CIC-IPN) for their invaluable collaboration in the publishing of this issue. The submission, review, and selection processes were enabled by the widely adopted tool EasyChair, (www.easychair.org) free of charge.

Juan Carlos Chimal Eguía
Elizabeth López Lozada
Federico Furlán Colín
CIC-IPN, Mexico
Guest Editors

October 2019

Table of Contents

	Page
Algoritmo esteganográfico basado en autómatas celulares y en la sustitución de los bits menos significativos	11
<i>Uriel López, Humberto Dávila, Luis Zapata, Marco Ramírez</i>	
Algoritmos híbridos paralelos para el problema de coloreado de grafos	21
<i>Jessica G. Urrea Guerrero, Xiomara P. Zaldivar Colado, Rolando Menchaca-Méndez</i>	
Análisis e identificación de locuciones verbales mexicanas	35
<i>Belém Priego Sánchez</i>	
Application of Natural Language Processing Techniques for Classification of Web Published News in Spanish	47
<i>Yadira Hernandez-Cruz, Angel Chi-Poot, Gilberto Martinez-Luna</i>	
Comparative Analysis of Interest Point Detectors Algorithms on Robotic Operative System.....	55
<i>Francisco Javier Valdepeña Rivera, Dante Mújica Vargas, Miguel Ángel Ruíz</i>	
Convolutional Model with Classification through Izhikevich Neuron	65
<i>Antonio Luna-Álvarez, Dante Mújica-Vargas, Manuel Mejía-Lavalle</i>	
Detección y rastreo de peatones empleando vectores de características de histogramas de gradientes orientados y patrones binarios locales en una máquina de soporte vectorial	81
<i>Raymundo Rodríguez Romero, Luis Pastor Sánchez Fernandez, Edgardo Manuel Felipe Riverón, José Juan Carbajal Hernández</i>	
Diseño e implementación de un sistema para determinar el impuesto predial a través del procesamiento y análisis de imágenes satelitales.....	95
<i>Roman Alejandro Hernández-Moreno, Jessica Gabriela Azpeitia-Rodríguez, Derlis Hernández-Lara, Leonardo Miguel Moreno-Villalba</i>	
Embedded System for Human Detection Applied to Domotics	103
<i>Oscar Arturo González González, Alina Mariana Pérez Soberanes, Víctor Hugo García Ortega, Julio César Sosa SAVEDRA</i>	

Estudio de herramientas para alcanzar niveles de calidad de software a nivel de proceso	117
<i>Yadira Jazmín Pérez Castillo, Sandra Dinora Orantes Jiménez</i>	
Evaluation of Five Classifiers for Depression Episodes Detection	129
<i>Susana L. Pacheco-González, Laura A. Zanella-Calzada, Carlos E. Galván-Tejada, Nubia M. Chávez-Lamas, J. Francisco Rivera-Gómez, Jorge I. Galván-Tejada</i>	
Generation of Non-Coordinate Navigation Knowledge from a Flow of Input Views	139
<i>Josué Figueroa-González, Georgii Khachaturov, Juan M. Martínez-Hernández</i>	
Image Encryption System Based on Cellular Automata and S-Box	153
<i>Juan Contreras, Marco Ramírez, Jesús Aboytes</i>	
MeIPEAS: An Intelligent Virtual Tutor for Mexican Elementary Schoolchildren	163
<i>Hector Rafael Orozco Aguirre, Gonzalo Ivan Riego Caravantes</i>	
Modelo difuso para la evaluación de condiciones apropiadas para turismo en litorales de México	179
<i>César A. Hoil-Rosas, José J. Carbajal-Hernández, Luis P. Sánchez-Fernández, Alberto D. Dávila-Lamas</i>	
Multilevel Image Transmission using Polar Codes	191
<i>Guadalupe Donaji Aguilar Licea, Saul Lazcano Salas</i>	
Prediction of Cancer using Microarrays Analysis by Machine Learning Algorithms	199
<i>José Luis Velázquez-Rodríguez, Yenny Villuendas-Rey, Cornelio Yáñez-Márquez</i>	
Procesador dedicado para el tratamiento de datos en formato de punto flotante	209
<i>Úrsula Samantha Morales-Rodríguez, Gelacio Castillo-Cabrera, Herón Molina-Lozano</i>	
Propuesta de sistema de gestión inteligente basado en IoT para hidroponía	219
<i>Evelyn Gutierrez Leon, Jorge Erik Montiel Arguijo, Chadwick Carreto Arellano, Felipe Rolando Menchaca García</i>	

Propuesta de sistema de gestión para optimización de redes de transporte público	235
<i>Jorge Erik Montiel Arguijo, Evelyn Gutiérrez León, Chadwick Carreto Arellano, Felipe Rolando Menchaca García</i>	
Prototipo de sistema embebido para la asistencia de personas con discapacidad visual en la acera pública	247
<i>Diego Cruz Mondragón, Katia Santillán Álvarez, Víctor Hugo García Ortega, Juan Humberto Sossa Azuela</i>	
Ramath: Mobile Application for Math Learning using Augmented Reality.....	261
<i>Hiram David Cortés Díaz, Octavio Elías Piáal Ramírez, Amadeo José Argüelles Cruz, Claudia Marina Vicario Solórzano</i>	
Selección óptima de cartera de inversión de stocks de S&P 500: Uso de algoritmos genéticos	271
<i>Marco Adrián Guido Torres, Volodymyr Ponomaryov</i>	
Sistema de monitoreo de consumo eléctrico con interfaz para teléfono inteligente	279
<i>Alejandro Andrés Serapio Carmona, Ismael Díaz Rangel, Rodolfo Zola García Lozano, Cuauhtémoc Hidalgo Cortés</i>	
Sistema de navegación y evasión de obstáculos aplicando un sistema de control difuso en una placa Arduino UNO	291
<i>David Jesus Segura Cristino, Heron Molina Lozano, Elsa Rubio Espino</i>	
Software de estación terrena para cohetes hidropulsados.....	305
<i>Alberto Carlos Pérez Cadena, Oswaldo Morales Matamoros, Diego Alfredo Padilla Pérez, Jesús Jaime Moreno Escobar</i>	
Teaching Basic Concepts: Geometric Forms and Colors on a NAO Robot Platform.....	323
<i>Diana Olvera, Uriel Escalona, Humberto Sossa</i>	
Transmission and Reception of Text and Images Using Visible Light.....	335
<i>Sergio Sandoval-Reyes</i>	
Voice Synthesis System Based on Recursive Functions Designed by Graphs	347
<i>Rodolfo Romero Herrera, José Elias Romero Martínez</i>	
When Twitter Becomes a Data Source for Geospatial Analysis.....	357
<i>Magdalena Saldana-Perez, Camille Cavalière, Miguel Torres-Ruiz, Marco Moreno-Ibarra</i>	

Exploring Storing Capacity of Hyperdimensional Binary Vectors..... 375
*Job Isaias Quiroz Mercado, Ricardo Barrón Fernández,
Marco Antonio Ramírez Salinas*

Context Pattern Based Agricultural Named Entity Recognition..... 383
Payal Biswas, Aditi Sharan, Ashish Kumar

Algoritmo esteganográfico basado en autómatas celulares y en la sustitución de los bits menos significativos

Uriel López, Humberto Dávila, Luis Zapata, Marco Ramírez

Universidad Autónoma de San Luis Potosí,
Coordinación Académica Región Altiplano Oeste, México
tulio.torres@uaslp.mx

Resumen. En esta investigación se presenta un nuevo algoritmo de esteganografía para imágenes, utilizando la sustitución de los bits menos significativo y la regla local 90 de autómatas celulares. Este algoritmo es capaz de cifrar y ocultar los píxeles de la imagen en un proceso combinado. El factor PSNR indica que no existe una distorsión severa en la imagen esteganográfica, lo que en esta área se interpreta como a menor distorsión de la imagen esteganográfica, mayor es la probabilidad de pasar desapercibida y que no sea analizada. Sin embargo, en caso de que la imagen oculta sea extraída, ésta se encuentra protegida por una clave secreta. Este algoritmo transforma la imagen secreta a una imagen cifrada con distribución uniforme y sin correlación. Este algoritmo esteganográfico puede ser una nueva opción para ocultar y proteger imágenes relacionando ambos procesos.

Palabras clave: esteganografía, autómatas celulares, PSNR.

Steganographic Algorithm Based on Cellular Automata and the Replacement of Less Significant Bits

Abstract. This research presents a new steganography algorithm for images, using the least significant bit replacement and local rule 90 of cellular automata. This algorithm is able to encrypt and hide the image pixels in a combined process. The PSNR factor indicates that there is no severe distortion in the steganographic image, which in this area is interpreted as the less distortion of the steganographic image, the greater the probability of going unnoticed and that it is not analyzed. However, if the hidden image is extracted, it is protected by a secret key. This algorithm transforms the secret image to an encrypted image with uniform distribution and no correlation. This steganographic algorithm can be a new option to hide and protect images by relating both processes.

Keywords: steganography, cellular automata, PSNR.

1. Introducción

En nuestra vida digital actual, hay una demanda latente por seguridad para garantizar que nuestra información confidencial no sea observada por personas sin autorización. Para ello se han desarrollado diversas técnicas, resolviendo diferentes aspectos de este problema. La esteganografía es una de las metodologías con las que se cuenta para ocultar y proteger nuestra información en algún objeto llamado portador. El objetivo es que la información pase inadvertida estando oculta en el portador. La naturaleza de los datos a esconder y el objeto portador puede no ser la misma, es decir, podemos ocultar un archivo de audio en una imagen, un texto en un audio etc.

Existen diferentes técnicas para aplicar la esteganografía, una de las más conocidas es la sustitución de los bits menos significativos. Esta técnica consiste en reemplazar los bits menos significativos del portador por los bits del archivo secreto. Por lo tanto, en el caso de que el portador y la información secreta sean imágenes, la percepción de la imagen portadora puede ser distinta cuando se monte la imagen secreta, incluso puede revelar ciertos patrones de ésta. La imagen resultante de sobreponer en la imagen portadora la imagen secreta, se le llama imagen esteganográfica.

Para evitar que la información sea simplemente revelada al momento de ser extraída del portador, varios esquemas han propuesto cifrar la información secreta para después ocultarla en el portador [1-3] haciendo dos procesos independientes. Dado que las imágenes digitales pueden tener una gran cantidad de datos, una alta correlación adyacente [4] o pueden ser totalmente redundantes; los algoritmos de cifrado suelen tener varias operaciones de confusión y difusión e iteraciones para poder romper la correlación adyacente. Lo que incrementa el costo computacional.

Es por eso que en esta investigación, se ha desarrollado un algoritmo que relaciona ambos procesos: criptografía y esteganografía; capaz de sustituir los valores de la imagen secreta con influencia de los bits de la imagen portadora y una llave inicial para lograr una mayor dispersión de los valores, y al mismo tiempo ocultar la imagen secreta. Gracias a la regla local 90 de autómatas celulares, con operaciones simples se logra una gran dinámica capaz de cifrar imágenes altamente redundantes, para obtener imágenes con histograma uniforme. El algoritmo está basado en un sistema de encriptación llamado ESAC (Encriptación basada en la sincronización de autómatas celulares) en la versión que se muestra en la Ref. [5], con algunas modificaciones y mejoras, que pueden hacer de este algoritmo una nueva propuesta para el área de esteganografía.

2. Materiales y métodos

2.1. Autómatas celulares

Un autómata celular es un modelo matemático para un sistema dinámico, que consiste en una red organizada de celdas que adquieren distintos estados o valores. Evolucionan a pasos discretos cambiando su estado, de acuerdo a un algoritmo llamado regla local, donde intervienen los valores de la celda y los de sus vecinos.

La regla local 90 es clasificada como de clase III [6], lo que quiere decir que es de patrón caótico, por lo que ha sido utilizada para crear generadores de números pseudoaleatorios y criptosistemas [5]. Es una de las reglas de los autómatas celulares

elementales, para su cálculo se considera una vecindad de radio uno y dos estados $\{0,1\}$, puede ser descrita con la siguiente expresión:

$$x_i^{t+1} = A(x_{i-1}^t + x_{i+1}^t), \quad (1)$$

donde t representa el tiempo e i la posición del autómata. Por lo tanto, el cálculo de esta regla equivale a calcular una operación XOR entre las celdas que están a los extremos de la vecindad.

2.2. Generador de números pseudoaleatorios y función de procesamiento

En la Ref. [7] se utilizó la regla local 90 y el fenómeno de sincronización que ésta presenta, para proponer un sistema de cifrado de datos llamado ESAC y un Generador de Números Pseudo-Aleatorios (GNPA). El GNPA funciona introduciendo dos vectores booleanos \mathbf{x} y \mathbf{y} de longitud n y $n+1$ bits respectivamente, para obtener la secuencia aleatoria \mathbf{t} , utilizando la función h que es con la que se denomina a la evolución hacia atrás del autómata celular. La Fig. 1 presenta un diagrama a bloques del GNPA.

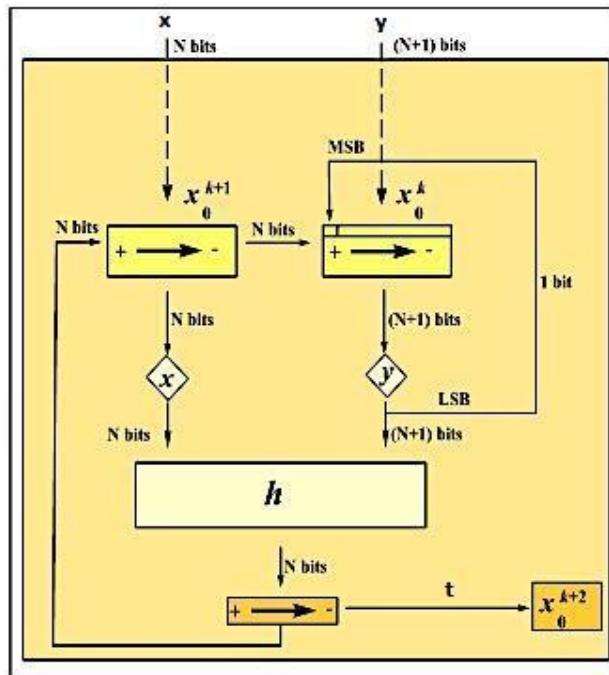


Fig. 1. Forma básica del generador de números pseudo-aleatorios.

El sistema se retroalimenta y la secuencia generada \mathbf{t} ocupa el lugar del vector \mathbf{x} , y la anterior \mathbf{x} toma el lugar de \mathbf{y} , para lo cual de la anterior versión se toma el bit menos significativo y se concatena como el bit más significativo. En la Ref. [8] se evaluó su

desempeño pasando todas las pruebas de NIST (National Institute of Standards and Technology).

En la Ref. [5] este generador se modificó para diseñar una función de procesamiento, capaz de intercambiar los coeficientes de pixeles idénticos, por valores diferentes en función del tiempo. Es totalmente reversible gracias a la regla local 90 y a las retroalimentaciones que utiliza. En este caso, el lugar del vector \mathbf{x} , es utilizado por los bloques en claro \mathbf{m} , y el vector \mathbf{y} es llamado \mathbf{z} , para diferenciar su uso. El bloque resultante se llama $\hat{\mathbf{m}}$.

Después de la primera iteración, \mathbf{z} se calcula con la retroalimentación, utilizando el bloque $\hat{\mathbf{m}}$ resultante, se le concatena el bit menos significativo del vector \mathbf{z} anterior. De esta manera aunque los bloques de texto en claro m_1 y m_2 sean iguales, son procesados con vectores \mathbf{z} diferentes, resultando \hat{m}_1 y \hat{m}_2 diferentes entre sí.

Esta función puede ser simplificada a ecuaciones de un solo tiempo, sin la necesidad de evolucionar el autómata, utilizando lógica booleana. Las ecuaciones (2) resultantes se muestran a continuación, para el caso de 8 bits.

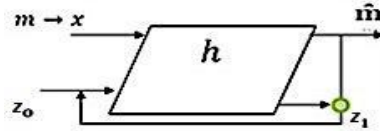


Fig. 2. Función de procesamiento, utilizando el generador de números pseudoaleatorios.

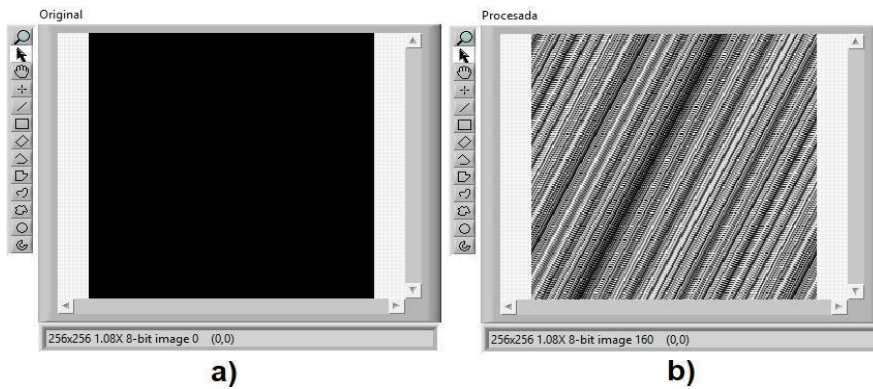


Fig. 3. Caso particular a mejorar: a) Imagen sólida con todos los pixeles $\mathbf{m}=\mathbf{0}$, b) Imagen sólida procesada.

$$\begin{aligned}
 \hat{m}_1 &= m_1 \oplus z_2, \\
 \hat{m}_2 &= m_2 \oplus z_1 \oplus z_3, \\
 \hat{m}_3 &= m_1 \oplus m_3 \oplus z_4, \\
 \hat{m}_4 &= m_4 \oplus z_1 \oplus z_3 \oplus z_5, \\
 \hat{m}_5 &= m_1 \oplus m_3 \oplus m_5 \oplus z_2 \oplus z_6, \\
 \hat{m}_6 &= m_2 \oplus m_6 \oplus z_1 \oplus z_5 \oplus z_7, \\
 \hat{m}_7 &= m_1 \oplus m_5 \oplus m_7 \oplus z_8, \\
 \hat{m}_8 &= m_8 \oplus z_1 \oplus z_5 \oplus z_7 \oplus z_9,
 \end{aligned} \tag{2}$$

donde \oplus representa la operación XOR.

Uno de los aspectos a mejorar de esta función, es el hecho de que no todos los bits se combinan de igual manera como lo hacen los más significativos, donde las ecuaciones involucran más bits. Este detalle se puede ver en el caso de una imagen sólida donde todos los píxeles son igual a cero, $\mathbf{m}=\mathbf{0}$. La versión procesada de esta imagen se muestra en la Fig. 3.

2.3. Método propuesto

Para implementar el sistema esteganográfico se utilizó el lenguaje de programación LabVIEW de National Instruments, trabajando con imágenes de 8 bits. Para solventar los problemas antes vistos en la función de procesamiento, se realizaron cambios en la conformación de los bloques, donde se hicieron pruebas ocultando 2, 3 y 4 bits en la imagen portadora, nombrando a cada versión como V1, V2 y V3 respectivamente. La imagen portadora debe ser de dimensiones mayores a la imagen oculta, para que ésta no pierda información, es decir si se almacenan dos bits de la imagen oculta, la portadora debe ser 4 veces más grande que la imagen a ocultar.

El método propuesto de esteganografía para el caso de 2 bits se calcula de la siguiente manera:

- 1) El bloque a procesar \mathbf{m} de 8 bits se conforma con los 6 bits más significativos de la imagen portadora, puestos en las posiciones de m_1 a m_6 . Los bits m_7 y m_8 son ocupados por bits de la imagen oculta.
- 2) Se genera una secuencia pseudoaleatoria \mathbf{z} , con los vectores \mathbf{x} y \mathbf{y} .
- 3) Se procesan el bloque \mathbf{m} y el vector \mathbf{z} .
- 4) Al bloque $\hat{\mathbf{m}}$ resultante se le extraen los bits \hat{m}_7 y \hat{m}_8 .
- 5) Los bits \hat{m}_7 y \hat{m}_8 se concatenan a los bits del pixel de la imagen portadora en la posición de los bits menos significativos, formando así un pixel de la imagen esteganográfica.
- 6) Se repite el proceso para el siguiente par de bits de la imagen oculta.

En la Fig. 4 se puede ver un diagrama de este método.

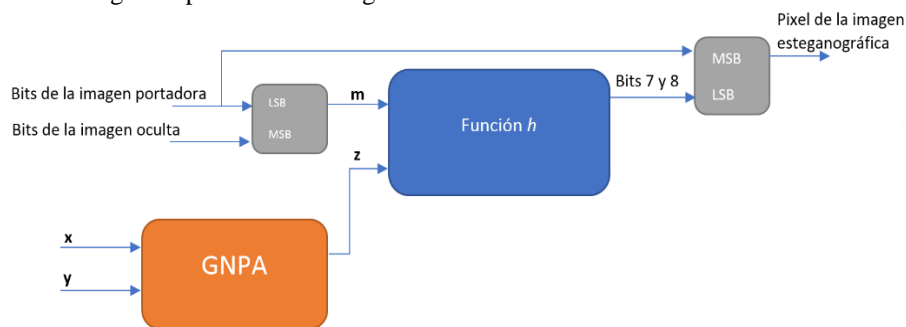


Fig. 4. Diagrama de bloques del algoritmo esteganográfico propuesto.

3. Resultados

A continuación, presentamos el análisis de las imágenes esteganográficas obtenidas, utilizando las diferentes versiones V1, V2 y V3. También se realizó un análisis

estadístico de las imágenes ocultas en su versión procesada, donde se obtuvieron los histogramas y coeficiente de correlación.

Estas pruebas se realizaron para diferentes tipos de imagen, las cuales son ampliamente utilizadas en el procesamiento de imágenes como lo son: Lena, mandril y pimientos, como imágenes ocultas de dimensiones de 256×256 píxeles a 8 bits. Como imagen portadora se utilizó la imagen de avión a 8 bits y de dimensiones de 512×512 para el caso de ocultar 2 bits por píxel, 768×256 para el caso de 3 bits por píxel, y por último de dimensiones 512×256 para el caso de 4 bits. La Fig. 5 muestra las imágenes de prueba.



Fig. 5. Imágenes de prueba utilizadas.



Fig. 6. a) Imagen portadora, b) Imagen esteganográfica con la imagen de Lena oculta.

3.1. PSNR

La métrica PSNR es utilizada para medir la calidad de imágenes tras un procesamiento. Para el caso de esteganografía, mientras mayor sea este valor menor es

la probabilidad de que por inspección visual se realice un ataque [9]. En esta investigación se utiliza para medir la calidad de la imagen esteganográfica, debido a que se suele distorsionar cuando se monta la imagen oculta, lo que puede levantar sospechas y que la imagen sea sustraída. La Fig. 6 muestra la imagen portadora utilizada y la imagen esteganográfica resultante, ocultando dos bits por pixel de la imagen de Lena.

La Tabla 1 muestra los valores de PSNR obtenidos para cada imagen en las diferentes versiones del algoritmo

En otras investigaciones [10-12] obtienen valores de PSNR mayores a los reportados, pero es muy probable que sea por las dimensiones de imágenes que se utilizan. En estas investigaciones la imagen portadora tiene una capacidad en bits mayor que la mínima requerida para ocultar la imagen secreta. En esta investigación siempre se utilizó la mínima capacidad de bits en la imagen portadora, por lo tanto, todos sus pixeles contienen datos de la imagen secreta.

Por otra parte, la imagen oculta resultante del procesamiento se puede ver en la Fig. 7, donde se ilustra el caso de una imagen sólida en color negro, con todos sus coeficientes igual a $m=0$.

Tabla 1. Cálculo del PSNR de la imagen esteganográfica en las diferentes versiones del algoritmo, con diferentes imágenes ocultas.

Imagen	V1 (dB)	V2 (dB)	V3 (dB)
Lena	41.4259	36.4046	29.2256
Mandrill	41.4327	36.3845	29.8222
Pimientos	41.4135	36.4111	29.6882

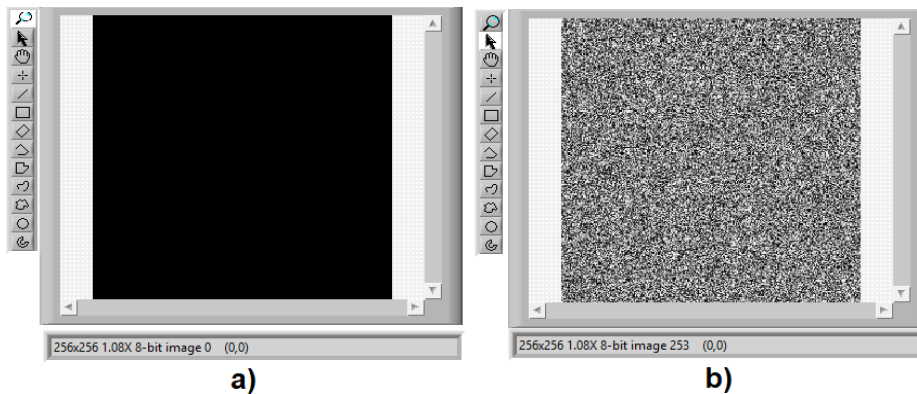


Fig. 7. Resultado de la nueva función de procesamiento, a) Imagen original sólida y b) Versión procesada de a).

Como podemos ver, ahora no muestra patrones, evoluciona de una manera más uniforme.

3.2 Análisis de histogramas

El análisis de histogramas revela la frecuencia con la que se presentan los valores de los coeficientes de los pixeles de una imagen. Por lo tanto, una imagen cifrada debe

presentar un histograma uniforme, lo que indica que logró ocultar la redundancia de la imagen original, que puede usarse como estadística para revelar información de la imagen oculta. En este análisis, todas las imágenes de prueba que se procesaron presentaron un histograma uniforme para todos los valores de la codificación, en este caso 8 bits. La Fig. 8 muestra el histograma de la imagen de los pimientos y su versión procesada.

3.3 Correlación

El cálculo de correlación nos ayuda a determinar si una imagen cifrada es independiente de su imagen original. Utilizamos el cálculo del coeficiente de correlación de Pearson, el cual señala que si el valor del índice es menor a $|0.3|$ no existe una correlación entre las señales [13]. La Tabla 2 muestra el resultado de calcular el coeficiente de correlación a las imágenes procesadas con su respectiva imagen original, para todas las versiones.

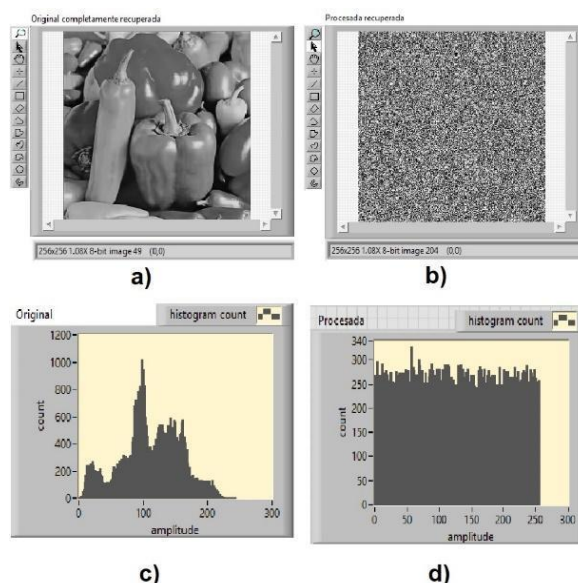


Fig. 8. Análisis de histogramas. a) Imagen original, b) Versión procesada de a), c) histograma de a) y d) histograma de b).

Tabla 2. Cálculo de correlación de las imágenes ocultas procesadas.

Imagen	V1	V2	V3
Lena	-0.0002887	-0.0021166	-0.0019959
Mandrill	0.0015039	-0.0002760	0.0004766
Pimientos	-0.0032853	-0.0005435	-0.0015039

Como podemos ver no existe correlación entre las imágenes, lo cual nos indica que no hay una dependencia entre las imágenes procesadas y su versión original.

4. Conclusiones

En este trabajo se presentó una propuesta para un algoritmo esteganográfico capaz de esconder y cifrar la imagen oculta. El utilizar la sincronización de autómatas celulares con los bits de la imagen portadora y el generador de números pseudoaleatorios, proporcionó una dinámica mayor a la de la función de procesamiento, logrando trabajar con imágenes altamente redundantes. La imagen procesada pasó un análisis estadístico básico y la imagen portadora se mantuvo en los límites de calidad. Por lo que podemos concluir que esta propuesta puede representar una opción viable para aplicaciones de seguridad. La regla local 90 para autómatas celulares brinda una dinámica interesante para aplicaciones de esteganografía debido a su comportamiento y a que es reversible.

A futuro esperamos estudiar más a fondo de este algoritmo y de ser necesario mejorar el sistema. Una de las metas es lograr que con el fenómeno de sincronización la imagen oculta tienda a ser similar a la imagen portadora, para que al momento de ser extraída, no se visualice como una imagen cifrada, sino como una imagen similar a la portadora.

Referencias

1. Gupta, S., Goyal, A., Bhushan, B.: Information hiding using least significant bit steganography and cryptography. *International Journal of Modern Education and Computer Science* 4(6), 27 (2012)
2. Chauhan, S., Kumar, J., Doegar, A.: Multiple layer text security using variable block size cryptography and image steganography. In: 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICIT), pp. 1–7, IEEE (2017)
3. Joshi, K., Yadav, R.: A new LSB-S image steganography method blend with Cryptography for secret communication. In: 2015 Third International Conference on Image Information Processing (ICIIP), pp. 86–90, IEEE (2015)
4. Lian, S.: *Multimedia content encryption: techniques and applications*. Auerbach Publications (2008)
5. Ramirez-Torres, M.T., Murguía, J.S., Carlos, M.M.: Image encryption with an improved cryptosystem based on a matrix approach. *International Journal of Modern Physics C*, 25(10), 1450054 (2014)
6. Wolfram, S.: *A new kind of science*. Vol. 5, p. 130, Champaign, IL: Wolfram media (2002)
7. Urias, J., Ugalde, E., Salazar, G.: A cryptosystem based on cellular automata. *Chaos*, Woodbury, NY, 8(4), 819–822 (1998)
8. Murguía, J.S., Carlos, M.M., Rosu, H.C., Flores-Eraña, G.: Improvement and analysis of a pseudo-random bit generator by means of cellular automata. *International Journal of Modern Physics C*, 21(06), 741–756 (2010)
9. Karim, S.M., Rahman, M.S., Hossain, M.I.: A new approach for LSB based image steganography using secret key. In: 14th international conference on computer and information technology (ICCIT 2011), pp. 286–291, IEEE (2011)
10. Jana, B., Giri, D., Mondal, S.K., Pal, P.: Image steganography based on cellular automata. *International Journal of Pure and Applied Mathematics* 83(5), 701–715 (2013)
11. Solís, I.S.: Esteganografía en imágenes digitales aplicando autómatas celulares bidimensionales como generadores pseudoaleatorios. *Revista de Investigaciones (Puno)-Escuela de Posgrado de la UNA PUNO*, 6(1), 66–77 (2017)
12. Bhardwaj, R., Sharma, V.: Image steganography based on complemented message and inverted bit LSB substitution. *Procedia Computer Science* 93, 832–838 (2016)

Uriel López, Humberto Dávila, Luis Zapata, Marco Ramírez

13. Mukaka, M.M.: A guide to appropriate use of correlation coefficient in medical research. *Malawi Medical Journal* 24(3), 69–71 (2012)

Algoritmos híbridos paralelos para el problema de coloreado de grafos

Jessica G. Urrea Guerrero¹, Xiomara P. Zaldivar Colado¹,
Rolando Menchaca-Méndez²

¹ Universidad Autónoma de Sinaloa, Culiacán, México

² Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México

jess.urreag@gmail.com, xiomara.zaldivar@uas.edu.mx,
rmen@cic.ipn.mx

Resumen El problema de coloreado de grafos pertenece a la clase de problemas *NP-Difícil*, por lo que a menos que la clase de problemas P sea igual a la clase NP , no existe un algoritmo polinomial que lo resuelva. A pesar de su elevada complejidad, el desarrollo de algoritmos para este tipo de problemas son de suma importancia por el hecho de que los problemas *NP-Difícil* aparecen en casi todas las áreas de las ciencias de la computación como aprendizaje de máquina, las redes de computadoras, los sistemas operativos, sistemas de información, entre otras. En este trabajo proponemos una implementación paralela, basada en CUDA, de un conjunto de variantes de un algoritmo evolutivo híbrido para el problema de coloreado de grafos. Dicho algoritmo se caracteriza por mezclar técnicas de cómputo evolutivo con heurísticas de búsqueda local, con el objetivo de encontrar soluciones cercanas a la óptima. Los resultados experimentales muestran que nuestra propuesta de implementación paralela es capaz de consistentemente encontrar buenas soluciones y al mismo tiempo reducir el tiempo de ejecución de los algoritmos evolutivos tradicionales.

Keywords: NP-difícil, coloreado de grafos, optimización combinatoria, metaheurísticas, algoritmos paralelos, CUDA.

Parallel Hybrid Algorithms for the Problem of Graph Coloring

Abstract. The graph coloring problem belongs to the problem class *NP-Hard*, so unless the problem class P is equal to the class NP , there is no polynomial algorithm to solve it. Despite its high complexity, the development of algorithms for these types of problems are of the utmost importance due to the fact that the problems *NP-Hard* appear in almost all areas of computer science as machine learning, computer networks, operating systems, information systems, among others. In this paper, we propose a parallel implementation, based on CUDA, of a set of variants of a hybrid evolutionary algorithm for the problem of coloring graphs. This

algorithm is characterized by mixing evolutionary computation techniques with local search heuristics, with the aim of finding near-optimal solutions. Experimental results show that our parallel implementation proposal is able to consistently find good solutions and at the same time reduce the execution time of traditional evolutionary algorithms.

Keywords: NP-hard, colored graphs, combinatorial optimization, metaheuristics, parallel algorithms, CUDA.

1. Introducción

La versión de decisión del problema de coloreado de grafos es uno de los 21 problemas que Richard Karp [1] demostró que pertenecen a la clase NP-Completa. Los problemas que pertenecen a esta clase se caracterizan por su complejidad, y por el hecho de que un algoritmo que resuelva cualquiera de ellos, podría ser usado para resolver a todos los demás problemas que pertenecen a dicha clase [1].

Existe otra clase de problemas conocida como NP-Difícil [2,3] que está compuesta por problemas aún más difíciles que los problemas NP-Completos. La complejidad extra radica en el hecho de que a diferencia de los problemas NP-Completos, los problemas NP-Difícil no se pueden verificar en tiempo polinomial.

Dado que el problema de coloreado de grafos en su versión de búsqueda pertenece a la clase de problemas NP-Difícil, es muy poco probable que exista un algoritmo de complejidad polinomial que lo resuelva. Por lo anterior, el problema ha sido abordado utilizando una gran variedad de técnicas algorítmicas entre las que se encuentran los algoritmos metaheurísticos. Una alternativa particularmente prometedora consiste en utilizar *algoritmos evolutivos híbridos* [4,5,6,9,10] que combinan algoritmos evolutivos tradicionales con esquemas de búsqueda local. En la literatura se muestra que este tipo de herramientas han dado buenos resultados [7], sin embargo, estos algoritmos suelen ser extremadamente costosos en términos de su tiempo de ejecución, por lo que se propone desarrollar una versión paralela que utilice la tecnología de tarjetas CUDA, con las cuales se espera una mejora significativa.

El resto del presente artículo se organiza de la siguiente manera. En la Sección 2 presentamos una muestra de los trabajos que proponen algoritmos para el problema de coloreado de grafos. En la Sección 3 formulamos el problema de coloreados de grafos como un problema de minimización del número de aristas monocromáticas del grafo. En la Sección 4 presentamos la especificación del algoritmo híbrido paralelo propuesto. Los resultados experimentales que caracterizan el desempeño del algoritmo propuesto en términos de la calidad de las soluciones y el tiempo de ejecución se presentan en la Sección 5 y finalmente, en la Sección 6 se presentan las conclusiones finales y los trabajos futuros.

2. Algoritmos metaheurísticos para coloreado de grafos

2.1. Búsqueda de vecindario variable

La búsqueda de vecindario variable o *variable neighborhood search* (VNS), se basa en la idea de un cambio dinámico del vecindario dentro de la búsqueda local. Procede por una metodología descendente que a través del vecindario hacia la exploración del óptimo local. Este método se mueve de la solución actual hacia una nueva, si es que implica una mejora en ella. De esta manera, la variable óptima se mantiene obteniendo vecindarios prometedores [11].

Avanthay [12] en 2003 introdujo una adaptación del método VNS para el problema de coloreado de grafos. Proponiendo una técnica que probó ser efectiva en determinar soluciones de alta calidad.

2.2. Algoritmos híbridos

Este tipo de algoritmos o técnicas son conocidos por combinar múltiples estrategias, ya sean del mismo tipo o diferentes para resolver un problema, también son llamados algoritmos *meméticos*.

Galinier [4] en 1999 realiza una propuesta utilizando un algoritmo híbrido para el problema de coloreado de grafos, combinando nuevas clases de operadores de cruce especializadas. Los resultados experimentales demostraron que el algoritmo de coloreado híbrido garantiza una mejora en los resultados dentro de su operador de búsqueda tabú, demostrando la importancia de la cruce dentro del proceso de la búsqueda. Además de realizar experimentos acerca del comportamiento del algoritmo referente a la evolución de la función de evaluación, así como la diversidad de la población. Al principio es evidente la necesidad de preservar la diversidad para que la búsqueda sea eficiente. Concluyendo con la afirmación del gran potencial que existe al utilizar este tipo de algoritmos para problemas combinatorios.

Mientras que en 2010 Zhipeng [13], propuso un algoritmo de metaheurística híbrida la cual integra búsqueda tabú dentro de un algoritmo evolutivo para resolver el problema de coloreado de grafos. En su trabajo destaca la importancia de la diversidad de los individuos. Utilizó una técnica *greedy* para cruzar a los individuos llamada *greedy partition crossover* (GPX), combinándolo con un operador de cruce multi-padre adaptativo, el cual demuestra obtener mejores resultados que con un operador de cruce de dos padres, además agregó una nueva función de "calificación de bondad" que considera la calidad y la diversidad de los individuos.

2.3. Algoritmos paralelos

El procesamiento paralelo es definido como ejecución concurrente o simultánea de instrucciones en una computadora. La motivación obvia de la paralelización es incrementar la eficiencia del procesamiento. Existen múltiples aplicaciones que requieren de grandes cantidades de procesamiento y análisis.

A pesar de que el uso del paralelismo es un tema reciente con tarjetas gráficas, en 1994 Lewandowski [14] introdujo el paralelismo en su trabajo por medio de una computadora paralela y un algoritmo híbrido para el coloreado de grafos, el cual combina una versión paralela del algoritmo Morgenstern *S-Impasse*, con una búsqueda exhaustiva *Branch and bound*. Lewandowski menciona que los resultados obtenidos pueden ser comparados con dos heurísticas secuenciales simples: el algoritmo de saturación de Brélaz (Dsat) y el algoritmo de Leighton RFL (*Recursive Largest First*). En general el algoritmo propone que un gestor de procesos empiece realizando la búsqueda exhaustiva paralelizada dentro de un grupo de procesadores, mientras que en otro grupo en paralelo se realice la heurística *S-Impasse*.

Por otro lado, en 2007 Szymon [15] propuso otro algoritmo paralelo con recocido simulado para el problema de coloreado de grafos, utilizando múltiples procesadores trabajando de forma concurrente en cadenas individuales con el objetivo de minimizar la función de costo y guardar la mejor solución encontrada. La coordinación del algoritmo es por medio de un modelo esclavo-maestro en el cual un procesador es responsable de recolectar las soluciones, elegir la solución actual y distribuir la cantidad de unidades esclavas. Los experimentos revelaron que el rendimiento depende mucho en escoger adecuadamente el proceso de enfriamiento, así a la generación inicial con un coloreado que se ajuste al problema. Además de demuestra que el modelo utilizado dentro del algoritmo alcanza su punto óptimo, cuando el número de esclavos es relativamente pequeño.

3. Planteamiento del problema

En el contexto de la teoría de grafos, los objetos del problema de coloreado de grafos se representan por medio de un conjunto de vértices V , y las relaciones son modeladas por un conjunto de aristas E entre vértices. Para el problema de coloreado de grafos, dos nodos $u, v \in V$ tal que $(u, v) \in E$ no pueden pertenecer a la misma clase. Una forma de distinguir las clases es utilizando un conjunto de colores, y a la división en clases se le conoce como coloreado [16].

Formalmente, la versión de búsqueda del problema de coloreado de grafos se define por un grafo no dirigido $G = (V, E)$, un número de colores disponibles k y consiste en encontrar una función $f : V \rightarrow \{1, 2, 3, \dots, k\}$ tal que el tamaño del conjunto de *aristas monocromáticas* $|M|$ es minimizado, donde $M = \{(u, v) : (u, v) \in E \text{ y } f(u) = f(v)\}$, es decir, que el conjunto M contiene todos los pares de nodos adyacentes cuya función de asignación de color, sea la misma.

Existen dos variantes de este problema, en la primera el valor k (número de colores) es fijo y se desea minimizar el número de aristas monocromáticas. En la segunda versión, k es variable y se requiere encontrar el valor mínimo de k , tal que el número de aristas monocromáticas es igual a cero.

Debido a que el problema de coloreado de grafos pertenece a la clase NP-Difícil, a menos que la clase de problemas P sea igual a la NP, no es posible diseñar un algoritmo que se ejecute en tiempo polinomial capaz de resolverlo.

Por lo anterior, se han utilizado técnicas metaheurísticas [8,17,18] no polinomiales para intentar encontrar buenas soluciones.

En el caso particular de los algoritmos híbridos, el principal cuello de botella radica en que cada individuo realiza una etapa de búsqueda local de manera secuencial. Para resolver este problema, se propone paralelizar la búsqueda local para que todos los individuos puedan progresar al mismo tiempo. Reducir el tiempo de ejecución es en particular importante cuando las instancias son muy grandes, o en casos en los que se requieren resultados en el menor tiempo posible, como lo es en los problemas de planificación.

4. Algoritmo evolutivo híbrido paralelo

En este trabajo se propone paralelizar en la plataforma CUDA variantes del algoritmo implementado por Galinier [4], por lo anterior, se concluyó que debido a que la memoria disponible dentro de la tarjeta CUDA es limitada, el implementar la técnica de búsqueda tabú no es una opción viable debido al consumo excesivo de recursos de cómputo que esta implica. Por lo tanto se propone el utilizar dos técnicas distintas para implementar la búsqueda estocástica y caracterizar el impacto que puede llegar a tener el tipo de búsqueda dentro del algoritmo evolutivo. Las técnicas propuestas son: *ascenso de colina* y *metrópolis*.

La parte que se desea paralelizar dentro del algoritmo genético híbrido es la búsqueda local. En el artículo de Galinier realiza la búsqueda de cada uno de los individuos dentro de un número fijo de generaciones, causando un retardo considerable en el tiempo. Mientras que en este trabajo se realiza la búsqueda local de cada individuo de forma paralela (asíncrona). Mediante el uso de los hilos CUDA, cada hilo toma un individuo realizando su búsqueda independiente, mientras que para evitar que el algoritmo avance, se utilizó una barrera CUDA, la cual garantiza que el algoritmo continúe una vez que todos los hilos hayan terminado de ejecutarse y avanzar a la siguiente generación. La población resultante será compuesta por óptimos locales que conformarán la nueva generación de individuos. Lo que respecta a la condición de parada, se utilizan dos aspectos: un número fijo de generaciones o cuando tres generaciones den los mismos resultados.

Se puede observar en la Figura 1 la metodología propuesta del algoritmo genético híbrido paralelo, para resolver el problema de coloreado de grafos.

Además del algoritmo genético junto con las búsquedas locales, se propone analizar las distintas jerarquías de memoria de la plataforma CUDA (global, compartida y local) en conjunto con variaciones en el tamaño de la población, número de bloques, e hilos, para ambas estrategias de búsqueda local, mediante experimentos exhaustivos para caracterizar el desempeño de cada algoritmo en términos del número cromático, y el tiempo total de ejecución. Para esto se utilizarán tanto experimentos estandarizados (*benchmarks*), como instancias generadas de manera aleatoria.

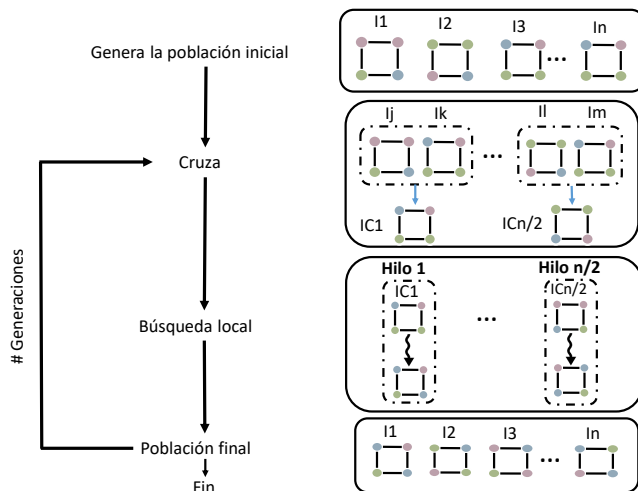


Fig. 1. Propuesta paralela para solucionar el problema de Coloreado de Grafos.

5. Resultados experimentales

Para la implementación de los experimentos se utilizó una computadora ASUS, con un procesador intel core i7 de 7^{ma} generación, la cual cuenta con una tarjeta gráfica GEFORCE GTX de NVIDIA, esta tarjeta contiene 768 núcleos CUDA, indispensables para la implementación en paralelo del problema de coloreado de grafos. Por otro lado, el algoritmo genético híbrido se codificó con python 3.6, junto con la extensión para CUDA en python llamada numba, con ella fue posible realizar la paralelización de python bajo ciertas restricciones, como lo fueron las estructuras aceptadas por numba dentro del kernel. Esto generó un cambio en la implementación del algoritmo paralelo respecto al secuencial. Lo que corresponde a las pruebas implementadas para el problema de coloreado de grafos, en este trabajo se realizó mediante el uso de dos tipos de experimentos: el primero fueron los grafos más conocidos para el problema de coloreado de grafos (benchmarks), y el segundo fueron grafos aleatorios generados con el modelo Gilbert G(n,p), en el que cada arista tiene una probabilidad independiente $0 < P < 1$ de aparecer en el grafo. Estos dos tipos de grafos fueron los utilizados en las pruebas bajo diferentes esquemas de configuraciones CUDA.

Las pruebas de los grafos aleatorios se realizaron en dos partes; la primera consistió en una comparación de las distintas configuraciones CUDA con ambas búsquedas locales, con los resultados obtenidos fue posible identificar a los representantes de cada búsqueda local. Una vez que se obtuvieron dichos representantes, en la segunda parte se analizó el comportamiento del problema dentro de diferentes esquemas, es decir, que se compararon los algoritmos greedy, el algoritmo genético híbrido secuencial y los representantes de las configuraciones

CUDA, todas ellas en base a los resultados de las pruebas con grafos aleatorios. Esto con el objetivo de analizar cual fue el algoritmo que brindó los mejores resultados.

5.1. Experimentos con grafos benchmarks

Los grafos más conocidos para el problema de coloreado (*benchmarks*) que fueron utilizados en las pruebas, son los que se muestran a continuación en la Tabla 1, así como los datos correspondientes de cada uno de ellos, como: el nombre del grafo, el número de nodos y el número cromático. Este último se refiere al menor número de colores encontrado hasta la fecha para colorear dichos grafos.

Tabla 1. Grafos *benchmarks* utilizados en pruebas.

Nombre	Num. Nodos	χ
myciel3	11	4
myciel4	23	5
2 - Fullins_3	52	5
Jean	80	10
Anna	138	11
DSJC250.5	250	28
DSJC500.5	500	48
le450_15c	450	15
le450_25c	450	25
flat300_28_300	300	31

El algoritmo genético híbrido paralelo diseñado para la plataforma CUDA, nos brindó la oportunidad de utilizar las diferentes jerarquías de memoria con las que cuenta la plataforma, dando la posibilidad de observar en que medida y como afecta la condición de competencia por los recursos, a medida que se va distanciando la memoria. Por lo anterior, se realizaron las pruebas correspondientes con cada búsqueda local tomando en cuenta el aspecto de la memoria, el número de hilos, y por último el número de bloques utilizados en el *kernel* CUDA, dependiendo del tamaño de la población (128 ó 256). En la Tabla 2 se observa como fueron organizadas estas configuraciones.

Tabla 2. Configuraciones del AGH-CUDA con respecto al tamaño de la población.

Población 128	Población 256
2 Bloques con 32 hilos	2 Bloques con 64 hilos
1 Bloque con 64 hilos	1 Bloque con 128 hilos

El motivo de utilizar diferentes tamaños de número de bloques y número de hilos, fue que en la literatura se menciona que la paralelización en CUDA tiende a dar buenos resultados si los hilos son múltiples de 32 en cada bloque, debido a

los warps (hilos en ejecución en el procesador). Experimentar con esto permitió analizar los diferentes resultados y el comportamiento del algoritmo. Por otro lado, lo que respecta al número de colores utilizado en el algoritmo paralelo, fue el mismo que con los otros algoritmos (número cromático χ), dando como resultado, la mejor configuración que fue capaz de encontrar, con el mínimo número de aristas monocromáticas.

Con el objetivo de poder realizar una comparación a profundidad sobre cada algoritmo, en las Tablas 3 y 4 se encuentran los resultados de los algoritmos *greedy*, el algoritmo genético híbrido secuencial, y los resultados del algoritmo genético híbrido paralelo con cada una de las diferentes configuraciones CUDA y búsquedas locales.

Tabla 3. Comparación de resultados de todas las diferentes estrategias para resolver el PCG.

	myciel3		myciel4		2-Fullin-3		Jean		Anna	
	Tiempo	NAM	Tiempo	NAM	Tiempo	NAM	Tiempo	NAM	Tiempo	NAM
smallest last	0.00082774	0	0.00141325	0	0.00390942	0	0.00752094	0	.01270669	0
Independent set	0.0015182	0	0.00352569	0	0.01781993	2.5	0.04130497	3.2	0.1002692	2.8
Secuencial-Metro-128	0.01187162	0	0.03661599	0	0.12785444	0	0.23786497	0	0.74481041	0
Secuencial-Metro-256	0.02434862	0	0.07471681	0	0.26140914	0	0.47922232	0	1.51863251	0
Secuencial-AC-128	0.0118885	0	0.03630571	0	0.13015926	0	0.24174876	0	0.757967	0
Secuencial-AC-256	0.02403321	0	0.07381628	0	0.26248481	0	0.47692354	0	1.50307961	0
Global.128.2.32 AC	0.01186512	0	0.03740296	0	0.13064239	0	0.24175212	0	0.75179014	0
Global.128.1.64 AC	0.01216662	0	0.03750648	0	0.12895861	0	0.23986328	0	0.75629468	0
Global.256.2.64 AC	0.02403724	0	0.07341001	0	0.2646903	0	0.48102677	0	1.50167754	0
Global.256.1.128 AC	0.02499306	0	0.07186005	0	0.24993515	0	0.45770319	0	1.4683805	0
Compartido, 128.2.32 AC	0.01167412	0	0.03760276	0	0.13293967	0	0.24412897	0	0.75360692	0
Compartido, 128.1.64 AC	0.01166618	0	0.03751254	0	0.13184466	0	0.24494262	0	0.74899721	0
Compartido, 256.2.64 AC	0.02343147	0	0.07029381	0	0.25618696	0	0.45926509	0	1.44184403	0
Compartido, 256.1.128 AC	0.02343309	0	0.06873243	0	0.24993711	0	0.48649487	0	1.4972954	0
Local.128.2.32 AC	0.01405582	0	0.03124785	0	0.12654028	0	0.22807615	0	0.7185735	0
Local.128.1.64 AC	0.01249394	0	0.03593137	0	0.13434041	0	0.23275118	0	0.72014003	0
Local.256.2.64 AC	0.02394104	0	0.07450573	0	0.25841138	0	0.47673178	0	1.51734428	0
Local.256.1.128 AC	0.02343206	0	0.07030921	0	0.25619063	0	0.46083097	0	1.52664618	0
Compartido, 128.1.64 AC	0.01166618	0	0.03751254	0	0.13184466	0	0.24494262	0	0.74899721	0
Global.128.2.32 Metro	0.0124676	0	0.03668928	0	0.1294549	0	0.24205575	0	0.76824338	0
Global.128.1.64 Metro	0.01208477	0	0.03680036	0	0.13353307	0	0.24884129	0	0.75777636	0
Global.256.2.64 Metro	0.02383668	0	0.0747021	0	0.25820274	0	0.47802968	0	1.49112074	0
Global.256.1.128 Metro	0.02343116	0	0.06561277	0	0.2483758	0	0.5311223	0	1.44964225	0
Compartido, 128.2.32 Metro	0.01245883	0	0.03904974	0	0.12496932	0	0.23276341	0	0.72482629	0
Compartido, 128.1.64 Metro	0.01217918	0	0.03811331	0	0.13223476	0	0.24504087	0	0.76275992	0
Compartido, 256.2.64 Metro	0.0241441	0	0.07400806	0	0.26549172	0	0.47712281	0	1.51675532	0
Compartido, 256.1.128 Metro	0.02424293	0	0.07539182	0	0.25961161	0	0.47422667	0	1.44497166	0
Local.128.2.32 Metro	0.01562073	0	0.03428657	0	0.12965605	0	0.23118894	0	0.72170324	0
Local.128.1.64 Metro	0.01249373	0	0.03593302	0	0.12805967	0	0.23275688	0	0.72170436	0
Local.256.2.64 Metro	0.02342687	0	0.06873183	0	0.25306246	0	0.45769961	0	1.43717144	0
Local.256.1.128 Metro	0.02414665	0	0.07439873	0	0.27027705	0	0.47712581	0	1.5648118	0

Gracias a las Tablas 3 y 4, es posible observar desde un panorama general cual fue el algoritmo que brindó los mejores resultados en ambos aspectos de evaluación (tiempo y NAM). El algoritmo genético secuencial con la búsqueda local de *gradient descent* y una población de 128, es el que demuestra sobresalir en ambos aspectos, se puede observar que hay una pequeña mejora en el número de aristas cuando la población aumenta a 256, pero los resultados en el NAM no representa una mejora en proporción al tamaño de la población, además de ser ejecutado en el doble de tiempo.

Tabla 4. Comparación de resultados de todas las diferentes estrategias para resolver el PCG (cont.).

	DSJC250.5		DSJC500.5		le450 15c		le450 25c		flat300 28	
	Tiempo	NAM	Tiempo	NAM	Tiempo	NAM	Tiempo	NAM	Tiempo	NAM
smallest last	0.144186	6042.8	0.57685297	25722.2	0.15067627	9309.2	0.16575511	3342.9	0.19865119	8394.2
Independent set	0.67416017	3264.3	4.24918582	13733.8	1.69596386	7320.3	2.27669196	3277	1.28648229	4735.8
Secuencial-Metro-128	34.0076456	127.6	211.857908	367.6	73.8724133	436.5	71.5198407	84.2	51.0609503	142.3
Secuencial-Metro-256	68.2377067	123.4	424.012502	365.3	144.784495	431.6	139.28404	81.8	97.6175751	138.6
Secuencial-AC-128	33.8509865	109	207.811258	346.7	73.6341539	412.1	73.747267	65.7	53.1880474	124.8
Secuencial-AC-256	70.7350266	107	443.754476	344.4	153.223014	411.5	145.053515	64.5	107.512961	123.7
Global, 128, 2, 32 AC	34.1717166	126.5	213.940944	367.6	72.9817449	433.4	73.1647449	83.3	51.7450323	139.3
Global, 128, 1, 64 AC	34.3502736	126.5	214.641463	365.1	73.133162	437.3	70.9769254	83.7	51.8578098	140.4
Global, 256, 2, 64 AC	64.92534	123.9	413.151073	362.5	139.837441	437.2	139.158007	81.4	99.3718972	138.8
Global, 256, 1, 128 AC	64.9549603	124	411.359283	364.6	139.243994	431.9	138.284741	81.6	97.7830749	137.3
Compartido, 128, 2, 32 AC	34.288105	126	216.863896	363.2	74.5060439	438	72.5581412	82.9	56.5781646	138.3
Compartido, 128, 1, 64 AC	34.5847636	124.2	213.416294	365.5	73.0850838	436.6	72.7328963	83.4	52.5269308	140.1
Compartido, 256, 2, 64 AC	68.1067633	124	427.984928	365	144.212103	433	138.417273	82.6	98.2642722	138.1
Compartido, 256, 1, 128 AC	67.4484771	121.7	429.068003	361.4	139.036223	434.1	142.060453	82.1	100.116905	136.3
Local, 128, 2, 32 AC	32.9921419	125	207.696505	364.8	70.207198	434.6	70.5585664	82.7	50.3506656	142.5
Local, 128, 1, 64 AC	33.0202579	125.2	204.747229	365.8	70.6045974	436.5	72.2004636	82	49.8178983	141.3
Local, 256, 2, 64 AC	67.0653446	123.4	430.594769	362.3	145.049938	431.2	148.267676	82.8	101.732809	138.4
Local, 256, 1, 128 AC	67.0821071	122.8	412.174914	363.9	141.39629	431.6	137.739474	82	98.5904665	138
Compartido, 128, 1, 64 AC	34.5847636	124.2	213.416294	365.5	73.0850838	436.6	72.7328963	83.4	52.5269308	140.1
Global, 128, 2, 32	33.7784859	126.2	211.159676	366.7	71.5544534	434.9	72.8595934	82.3	51.8439337	141.2
Global, 128, 1, 64	33.7664801	123.7	214.284681	365.6	70.9570144	437.9	71.6842514	82.8	52.8520553	140.2
Global, 256, 2, 64	66.5663851	123.5	421.985079	363.5	145.053013	437.5	149.404148	82.5	101.116071	140
Global, 256, 1, 128	65.5345762	123.7	411.801365	366	138.462901	433.1	139.13256	83.6	102.435103	138.6
Compartido, 128, 2, 32	32.2797088	125.5	205.553325	367.1	69.78218	434.8	69.8541036	81.8	54.7158808	139.6
Compartido, 128, 1, 64	33.6483535	125.1	213.496964	366.8	71.8928846	435.7	71.8097772	83.2	51.7925991	141
Compartido, 256, 2, 64	66.5423517	122	430.630632	360.9	141.605158	432.6	142.871373	82.6	100.438336	138.9
Compartido, 256, 1, 128	63.3256643	125.3	421.142926	362.3	142.19439	437	141.475995	80	100.910794	140.2
Local, 128, 2, 32	32.3939242	127	206.529582	363.1	70.2711495	434.9	69.1058066	82.8	48.9696568	139.9
Local, 128, 1, 64	32.3908168	127	207.107711	364.2	68.9168411	439.2	69.597785	82.9	48.9541056	141.6
Local, 256, 2, 64	64.4675533	123.4	415.066547	362	142.391495	432.7	136.831269	80.7	99.8263281	138.3
Local, 256, 1, 128	67.2900504	122.6	444.546514	361.6	144.741161	436.1	141.839243	82.2	99.9947874	139

Respecto a los resultados con el algoritmo genético paralelo, es posible observar que la configuración más cercana al algoritmo secuencial, es aquel que utiliza memoria local, con resultados que muestran una ligera mejora en el tiempo, pero con un mayor número de aristas monocromáticas. Por lo que se puede suponer que la memoria local es que es muy pequeña, y por lo tanto, es muy probable que realice varias copias de diferentes datos de la matriz, generando una competencia por los recursos al pasar por el bus, es decir que al existir esa competencia, esto hace que los distintos hilos se formen en el bus para obtener los datos a la memoria global, perdiendo por completo la paralelización del algoritmo.

Con el objetivo de medir la estabilidad de los algoritmos, se realizó el calculo de la desviación estándar de los 10 resultados obtenidos con cada una de las diferentes semillas, además del calculo de la desviación estándar de cada grafo, utilizando múltiples semillas.

5.2. Experimentos con grafos aleatorios

Además de los grafos *benchmark* también fueron creados 160 grafos aleatorios distintos. Para crear dichos grafos se propuso utilizar el modelo Gilbert $G(n,p)$, el cual funciona por medio del número de nodos del grafo (n) y una probabilidad (p), en la que cada arista tiene una probabilidad de que exista o no dentro del

grafo. En la Tabla 5 muestra el número de grafos que fueron creados, cada uno con probabilidades de 0.4 a 0.7 con variaciones en el número de nodos:

Tabla 5. Grafos aleatorios utilizados en pruebas.

Grafo	Probabilidad	Número de nodos
10 Grafos	0.4 - 0.7	20
10 Grafos	0.4 - 0.7	40
10 Grafos	0.4 - 0.7	60
10 Grafos	0.4 - 0.7	80
10 Grafos	0.4 - 0.7	100

Así como en los grafos *benchmark*, en los grafos aleatorios fueron utilizados las mismas estrategias de búsqueda local (*smallest last* e *independent set*), además de los mismos criterios de evaluación: el tiempo en segundos, y el número de aristas monocromáticas, éste último como con los grafos *benchmarks*, una vez que se fija el número de colores (k) en el algoritmo genético híbrido, los nodos cuyas etiquetas (colores) sean mayores a k , son sumados como aristas monocromáticas.

Los experimentos con grafos aleatorios se dividió en dos partes; la primera consistió en comparar las distintas configuraciones CUDA, para obtener representantes para cada una de las estrategias de búsqueda local. Para realizarlo, se tomó los resultados de los grafos de 20, 60 y 100 nodos, los resultados ganadores fueron aquellas que ganaban en dos de tres casos para cada estrategia, además de que fue dividida en las dos categorías de evaluación: tiempo y número de aristas monocromáticas. En la Tabla 6 muestra a los representantes ganadores.

Tabla 6. Representantes de las configuraciones CUDA.

Ascenso de colina		Metrópolis	
Criterio	Configuración CUDA	Criterio	Configuración CUDA
<i>Tiempo</i>	P128,1B,64H memoria local	<i>Tiempo</i>	P128,2B,32H memoria local
<i>NAM</i>	P256,1B,128H memoria global	<i>NAM</i>	P256,1B,128H memoria global

Una vez que se obtuvieron a los representantes de las configuraciones CUDA para cada búsqueda local, fue posible avanzar a la segunda parte de la experimentación, en la cual, se crearon gráficas que comparan los resultados del algoritmo *greedy*, secuencial, y los representantes de las configuraciones CUDA.

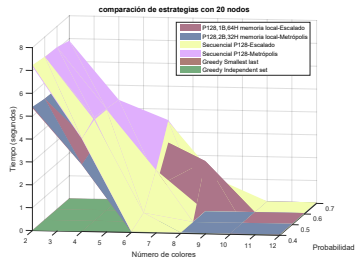


Fig. 2. Gráfica comparativa del tiempo de los algoritmos con 20 nodos.

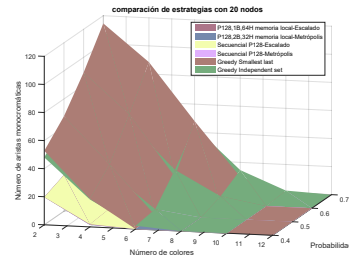


Fig. 3. Gráfica comparativa del NAM de los algoritmos con 20 nodos.

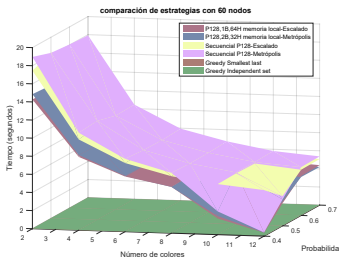


Fig. 4. Gráfica comparativa del tiempo de los algoritmos con 60 nodos.

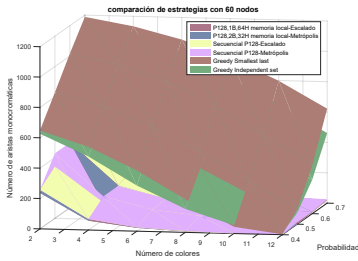


Fig. 5. Gráfica comparativa del NAM de los algoritmos con 60 nodos.

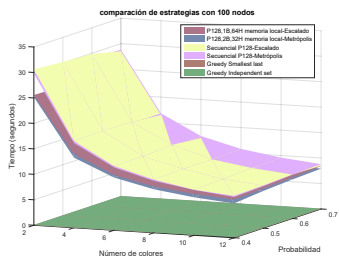


Fig. 6. Gráfica comparativa del tiempo de los algoritmos con 100 nodos.

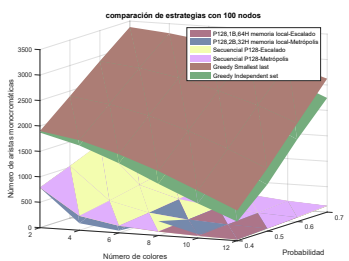


Fig. 7. Gráfica comparativa del NAM de los algoritmos con 100 nodos.

6. Conclusiones y trabajos a futuro

6.1. Conclusiones

Los resultados demostraron que con grafos *benchmarks*, el algoritmo secuencial es el que brinda el menor tiempo y menor número de aristas monocromáticas. La hipótesis resultante del análisis de estos resultados, es que existe un bajo

desempeño en la calidad de números aleatorios paralelos, a comparación con los generadores de números aleatorios secuenciales, por lo cual esto afecta la calidad de resultados de las estrategias de búsqueda local dentro del espacio de soluciones. Mientras que en los experimentos con grafos aleatorios, en la mayoría de los casos existe una clara superioridad de los algoritmos genéticos híbridos paralelos, demostrando la hipótesis inicial, que el paralelizar los algoritmos reduce el tiempo de ejecución.

Además se demuestra que los algoritmos evolutivos híbridos paralelos son capaces, en general, de encontrar mejores soluciones que los algoritmos voraces. Esto se debe, principalmente, a la capacidad de los algoritmos evolutivos de realizar una exploración más extensa del espacio de posibles soluciones, y de explotar las diferentes regiones del espacio de soluciones a partir de las búsquedas locales. Los algoritmos voraces, por su parte, tienden a quedar atrapados en óptimos locales que pueden ser arbitrariamente malos.

Con el propósito de caracterizar el impacto de incrementar el número de hilos, el tiempo de ejecución del algoritmo, así como la calidad de las soluciones al incrementar el tamaño de la población, se realizó un conjunto de experimentos donde varió el número de hilos desde 64 hasta 128.

Además de caracterizar el número de hilos, se analizó el impacto de utilizar diferentes esquemas de memoria de la tarjeta gráfica en el tiempo de ejecución del algoritmo. Los resultados muestran que la configuración con una *población de 256,1 bloque, 128 hilos con memoria global y metrópolis* fue la que presentó un mejor desempeño. Esto se debe principalmente a que una población grande asegura contener más puntos dentro del espacio de búsqueda, mientras que la búsqueda local metrópolis da la oportunidad de repartir los puntos, además de que la nobleza que brinda esta búsqueda permite no quedar estancados en óptimos locales, como encontrar nuevos caminos a posibles soluciones potenciales.

Es importante resaltar que debido a las condiciones de competencia generadas dependiendo de la distancia y limitación de la memoria, así como el uso constante de los buses de comunicación, no permitió observar una disminución del tiempo de ejecución del algoritmo propuesto de manera proporcional al número de hilos. Por esto último, es importante seguir investigando sobre nuevas implementaciones que hagan un uso más eficiente de la arquitectura CUDA con el fin de mejorar, aún más, el tiempo de ejecución.

6.2. Trabajo a futuro

Como trabajo a futuro se pretende aplicar la herramienta de software desarrollada para resolver problemas reales que pueden ser modelados como el problema de coloreado de grafos. En particular problemas en el contexto de planificación de tareas [19], planificación de grupos de trabajo [20], diseño de compiladores [21], reconocimiento de patrones [22], redes de computadoras [23,24].

Diseñar, implementar y caracterizar experimentalmente nuevas estrategias de búsqueda local específicamente diseñadas para plataformas de hardware con memoria restringida como las tarjetas CUDA. En este mismo sentido, es importante seguir investigando sobre algoritmos concurrentes que reduzcan las condiciones

de competencia que se presentan cuando varios núcleos de ejecución intentan acceder a la memoria compartida.

Extender la investigación reportada en la presente a otros problemas de optimización combinatoria que pertenezcan a la clase NP-difícil como cobertura de vértices, conjunto independiente, conjunto dominante, ciclo Hamiltoniano, entre otros.

Referencias

1. Karp, R.M., et al.: Reducibility among combinatorial problems. Complexity of computer computations, Springer, pp. 85–103 (1972)
2. Cormen, T.H.: Introduction to algorithms. MIT press (2009)
3. Kleinberg, J., Tardos, E.: Algorithm design. Pearson Education (2006)
4. Galinier, P., Hao, J.K.: Hybrid evolutionary algorithms for graph coloring. Journal of combinatorial optimization 3(4), 379–397 (1999)
5. Glover, F., Kelly, J.P., Laguna, M.: Genetic algorithms and tabu search: hybrids for optimization. Computers & Operations Research 22(1), 111–134 (1995)
6. Chiarandini, M., Stützle, T.: An application of iterated local search to graph coloring problem. In: Proceedings of the Computational Symposium on Graph Coloring and its Generalizations, pp. 112–125 (2002)
7. Krasnogor, N., Smith, J.: A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation 9(5), 474–488 (2005)
8. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR) 35(3), 268–308 (2003)
9. Beasley, D., Bull, D.R., Martin, R.R.: An overview of genetic algorithms: Part 1, fundamentals. University computing 15(2), 56–69 (1993)
10. Mühlenbein, H.: Parallel genetic algorithms, population genetics and combinatorial optimization. In: Workshop on Parallel Processing: Logic, Organization, and Technology, pp. 398–406. Springer, Berlin, Heidelberg (1989)
11. Baghel, M., Agrawal, S., Silakari, S.: Survey of metaheuristic algorithms for combinatorial optimization. International Journal of Computer Applications 58(19) (2012)
12. Avanthay, C., Hertz, A., Zufferey, N.: A Variable Neighborhood Search for Graph Coloring. Eur. J. Oper. Res., vol. 151, pp. 379–388 (2003)
13. Lü, Z., Hao, J.K.: A memetic algorithm for graph coloring. European Journal of Operational Research 203(1), 241–250 (2010)
14. Lewandowski, G., Condon, A.: Experiments with parallel graph coloring heuristics and applications of graph coloring. DIMACS Series in Discrete Mathematics (1994)
15. Łukasik, S., Kokosiński, Z., Świętoń, G.: Parallel simulated annealing algorithm for graph coloring problem. In: International Conference on Parallel Processing and Applied Mathematics. Springer, Berlin, Heidelberg, pp. 229–238 (2007)
16. Jensen, T.R., Toft, B.: Graph coloring problems (Vol. 39). John Wiley & Sons (2011)
17. Puchinger, J., Raidl, G.R.: Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: International Work-Conference on the Interplay Between Natural and Artificial Computation, Springer, Berlin, Heidelberg, pp. 41–53 (2005)

18. Voß, S., Martello, S., Osman, I.H., Roucairol, C. (eds.): Meta-heuristics: Advances and trends in local search paradigms for optimization. Springer Science & Business Media (2012)
19. Leighton, F.T.: A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards* 84(6), 489–506 (1979)
20. Gamache, M., Hertz, A., Ouellet, J.O.: A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding. *Computers and operations research* 34(8), 2384–2395 (2007)
21. Chaitin, G.J.: Register allocation and spilling via graph coloring. *ACM Sigplan Notices* 17(6) (1982)
22. Conte, D., et al.: Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence* 18(03), 265–298 (2004)
23. Zhu, X., Linglong, D., Zhaocheng, W.: Graph coloring based pilot allocation to mitigate pilot contamination for multi-cell massive MIMO systems. *IEEE Communications Letters* 19(10), 1842–1845 (2015)
24. Checco, A., Leith, D.J.: Fast, Responsive Decentralized Graph Coloring. *IEEE/ACM Transactions on Networking* 25(6), 3628–3640 (2017)

Análisis e identificación de locuciones verbales mexicanas

Belém Priego Sánchez

Universidad Autónoma Metropolitana unidad Azcapotzalco,
Departamento de Sistemas, CDMX, México
abps@azc.uam.mx

Resumen. Esta investigación presenta los resultados obtenidos tras analizar e identificar locuciones verbales en la variante diatópica general del español mexicano. Centrándose en el estudio morfosintáctico de estas locuciones, las cuales son un tipo específico de unidades fraseológicas, que se componen de un verbo y una o más partes de la oración (variables). El propósito de esta investigación es identificar automáticamente la existencia de locuciones verbales en textos planos (noticias), utilizando diferentes métodos supervisados de aprendizaje automático. Entre los hallazgos más relevantes es posible destacar que el clasificador J48 fue el que mejor desempeño demostró con un 76.74% de exactitud, superando a los métodos de clasificación probabilísticos y vectoriales. Los hallazgos demuestran que la existencia y uso de estas expresiones en textos escritos en español mexicano es frecuente; así también, comprueban que la identificación automática de locuciones verbales es una tarea lingüística-computacional compleja.

Palabras clave: unidades fraseológicas, locuciones verbales, identificación automática, métodos supervisados, aprendizaje automático.

Analysis and Identification of Mexican Verbal Phrases

Abstract. This research presents the results obtained after analyzing and identify verbal locutions in the general diatopic variant of Mexican Spanish. Focusing on the morphosyntactic study of these locutions, which are a specific type of phraseological units, which are composed of a verb and one or more parts of the sentence (variables). The purpose of this research is to automatically identify the existence of verbal phrases in plain texts (news), using different supervised methods of machine learning. Among the most relevant findings, it is possible to highlight that the J48 classifier showed the best performance with 76.74% accuracy, surpassing the probabilistic and vector classification methods. The findings show that the existence and use of these expressions in texts written in Mexican Spanish is frequent; likewise, they verify that the automatic

identification of verbal phrases is a complex linguistic-computational task.

Keywords: unidades fraseológicas, locuciones verbales, identificación automática, métodos supervisados, aprendizaje automático.

1. Introducción

La habilidad de la comunicación se desarrolla mediante información que se adquiere, capta y memoriza a lo largo de la vida. Dicha información es representada en forma de frases, expresiones, dichos, que permiten formular una idea o un concepto, y que son el campo de estudio de la fraseología, el cual es independiente de la lingüística que agrupa diversas disciplinas como la morfología, la sintaxis, la semántica, la pragmática, la sociolingüística y la psicolingüística [21]. La palabra “unidad” hace alusión a la propiedad, de estas expresiones, de actuar como un bloque y “fraseológica” se refiere a que están formadas por varias palabras [14]. éstas son unidades léxicas compuestas, o combinaciones de palabras, que poseen una amplia gama de construcciones lingüísticas como los modismos (tormenta en un vaso de agua, barrer bajo la alfombra), las expresiones fijas (*in vitro*, *por lo general*, *rock and roll*), sustantivos compuestos (*aceite de oliva*, *abrelatas*), y muchas más.

En la lingüística española se emplean diferentes variantes terminológicas que hacen referencia a las unidades fraseológicas tales como: “unidad fraseológica, fraseologismo, modismos, locuciones, expresiones fijas, dichos, frases hechas, expresiones idiomáticas, expresiones, frases, refranes, proverbios, modos de decir, aforismos”, entre otras. En [18] aparecen hasta 64 denominaciones diferentes. En este trabajo, se empleará el término “unidad fraseológica” debido a que goza de una gran aceptación en la filología española y es conocido en la fraseología internacional. Además, de que es el término más adoptado por la mayoría de los investigadores y aceptado por la lingüística soviética, pionera en el estudio y consolidación de la fraseología como disciplina científica [6].

Este trabajo de investigación está delimitado en analizar locuciones verbales, como objeto de estudio, para la variante diatópica general del español mexicano. Algunos ejemplos, que ilustran este tipo de expresiones, son los siguientes:

1. a) *colgar los tenis*
b) *acabarse la candela*
2. a) *zafarse un tornillo*
b) *perder el norte*

El ejemplo (1) significa morir y el segundo, perder la razón. Las expresiones (1a) y (2a) son locuciones verbales para el caso del español de México, mientras que (1b) y (2b) son locuciones verbales para el español de España; note que existe variabilidad léxica entre ambas variantes diatópicas del español. Como se nota en los ejemplos (1) y (2), éstos proporcionan un significado diferente al que los léxicos, separadamente, aportan. Este hecho nos remite a la no composicionalidad como rasgo propio de las unidades fraseológicas. Tal como menciona

González Rey [11] “el sentido idiomático de la expresión no reside, pues, en la no composicionalidad de sus componentes, sino en la no interpretación literal del enunciado”.

El estudio realizado, consiste en el análisis morfosintáctico de las locuciones verbales con el objetivo principal de identificar semi-automáticamente en un texto plano (noticias) si existe presencia o no de este tipo de estructuras lingüísticas, basándonos en la no composicionalidad. Tome en consideración el siguiente ejemplo:

3. *María colgó los tenis anoche.*

El ejemplo (3) tendría dos significados, uno literal y uno composicional, que dependen del contexto y utilización que se esté dando. Entonces la problemática reside en identificar de manera automática y computacional la interpretación de “colgar los tenis” en el contexto dado, es decir, determinar el sentido composicional de la expresión identificada.

2. Unidades fraseológicas

La fraseología, considerada como el reflejo del patrimonio cultural incluso folklórico de una comunidad lingüística, ha adquirido durante las últimas décadas un estado de investigación en la lingüística teórica [16]. Por mencionar algunos de los trabajos centrados en la fraseología española se puede encontrar un trabajo pionero presentado en [5], así como los trabajos relacionados [25,6,10]. Todos estos autores coinciden en que la unidad básica de análisis de la fraseología es la Unidad Fraseológica (de aquí en adelante UF), también llamada fraseologismo. Las Unidades Fraseológicas (denotadas por UFS) pertenecen a lo que en [7] se denomina “discurso repetido”, caracterizándose, sobre todo, por los tres rasgos siguientes:

- Su carácter poliléxico, es decir, que están formadas de al menos dos categorías gramaticales; característica que las distingue de las palabras simples.
- Su fijación, que implica su memorización como si constituyeran un todo inseparable, tal y como se almacenan las unidades simples.
- Su idiomatidad u opacidad léxica, al ser una agrupación de por lo menos dos palabras, su significado no es necesariamente deducido del significado de cada una de las palabras que la componen.

Para ejemplificar estas características considere la siguiente oración:

4. *Juan dejó con el ojo cuadro a María al regalarle flores.*

En el ejemplo (4), la expresión: “dejó con el ojo cuadrado” está formada de cinco categorías gramaticales, un verbo, una preposición, un artículo, un sustantivo y un adjetivo, ese es su carácter poliléxico. Si se cambiara la palabra “ojo” por otra parte de la cara, para los hablantes nativos del español mexicano,

sería una expresión totalmente diferente y posiblemente no entendida dado que la expresión es concebida y entendida como un todo y, en algunos casos, los elementos que la componen no se pueden modificar; a eso se refiere la fijación. El significado de la frase es “impresionar”, el cual no puede ser semánticamente inferido de los componentes léxicos “dejó”, “con”, “el”, “ojo” y “cuadrado”, justamente esa es la idiomatidad de la expresión.

Los autores [24,19], dedicados al estudio de las Variantes Fraseológicas (que denotaremos como VF), reiteran que las VF vienen prefijadas, es decir, que la variación está determinada y limitada, por lo que no se puede alterar y es conocida por los hablantes. Zuloaga [25] distingue dos tipos de variantes de las unidades fraseológicas (UF): las sinónimas y las que varían completamente su designación con la modificación de algunos de sus componentes. Ejemplos para la primera variación, son el (5) y el (6); mientras que, para el segundo grupo, el (7) y (8).

5. a) *llorar como una Magdalena*
b) *estar en un mar de lágrimas*
6. a) *estirar la pata*
b) *colgar los tenis*
c) *entregar el equipo*
7. a) *tener buena fama*
b) *tener mala fama*
8. a) *bajar el listón*
b) *subir el listón*

Para el caso del ejemplo (5), la unidad fraseológica (5a) es sinónima de la unidad fraseológica (5b), el significado es llorar demasiado. En la unidad fraseológica (6), el significado es morir y éstas (6a, 6b y 6c) son sinónimas. El ejemplo (7a), significa tener buena reputación, mientras que el (7b) lo contrario. En (8a) el significado es disminuir las exigencias y en (8b) aumentarlas.

Con este hecho, se nota que las unidades fraseológicas tienen cambios morfológicos, sintácticos y léxicos. Un cambio morfológico es en el cual uno de los componentes de la unidad fraseológica sufre alguna alteración. Estos cambios pueden ser de género, de número, de cuantificación, de determinación. Por ejemplo: (9) *más pobre que un perro* por (10) *más pobre que los perros*; (11) *romper en pedazos* por (12) *romper en mil pedazos*; (13) *ponerse gallo* por (14) *ponerse gallito*; (15) *andar de pata de perro* por (16) *andar de pata de perrito*; (17) *volver el agua al río* por (18) *volver las aguas a los ríos*. Las variantes sintácticas se dan cuando surgen cambios o alteraciones en el orden de los elementos de la UF, pero que no influyen en la lexicalización de la UF. Por ejemplo: (19) *mover cielo y tierra* por (20) *mover tierra y cielo*; (21) *del derecho y del revés* por (22) *del revés y del derecho*. Las variantes fraseológicas más frecuentes son aquellas que sustituyen un elemento léxico por otro. Por ejemplo: (23) *me importa un pepino* por (24) *me importa un comino* o por (25) *me importa rábano*. Por tanto, la problemática de las unidades fraseológicas viene del hecho que no son totalmente fijas [13,1,8,16]. Además de los cambios que las unidades fraseológicas tienen, existen diferentes clasificaciones de éstas.

Las clasificaciones de las unidades fraseológicas han surgido, la mayoría, como resultado de los problemas prácticos a los que se ha tenido que enfrentar el lexicógrafo al incluir la información fraseológica en la elaboración de diccionarios [6]. Las investigaciones sobre la fraseología española han surgido a comienzos de los años cincuenta con [5] y que dio la primera clasificación de las UFS. Posteriormente, en [7] se estableció la distinción entre la técnica libre del discurso y el discurso repetido. Trabajos basados en las investigaciones alemanas y soviéticas son los presentados en [22,25,15,3,4,23]. En el trabajo [22] se abordó la fraseología de las lenguas romances, mientras que en [25] se mejora y complementa la clasificación del trabajo presentado en [5]. Posteriormente, en el trabajo de [6] se propone una clasificación de las UFS a partir del trabajo [12], y además toma en cuenta características de los trabajos anteriormente mencionados; enseguida a esta clasificación, aparece la clasificación de [21].

2.1. Locuciones verbales

Las locuciones, son definidas en [5] como una “combinación estable de dos o más términos, que funciona como elemento oracional y cuyo sentido unitario consabido no se justifica, sin más, como una suma del significado normal de los componentes”. Las diferentes definiciones de locución en español han seguido esta caracterización y se han dividido según la función oracional que desempeñen, independientemente de que sean conmutables por palabras simples o por sintagmas. En el trabajo [6] se distinguen los siguientes tipos de locuciones: las nominales, las adjetivas, las adverbiales, las verbales, las prepositivas, las conjuntivas y las clausales.

Una Locución Verbal (denotada, de aquí en adelante, por LV) es una UF que contiene un verbo en el centro de su gramática. Desde el punto de vista sintáctico, expresan procesos y actúan como los predicados, con o sin complementos. Estas UFS, al igual que los verbos, se combinan con el sujeto y los complementos para formar una oración. En el trabajo [20] se define a una locución verbal como un grupo de palabras en las que al menos una es un verbo que funciona como núcleo del predicado, es decir, expresiones idiomáticas de significado no composicional. Son sintagmas fijos e idiomáticos cuya interpretación no se obtiene de la suma de sus partes, tomadas por separado. Definición que se considera a lo largo del desarrollo de esta investigación.

Las LV permiten ciertas modificaciones de orden sintagmático y/o paradigmático creando también los grados de fijación [13]. Por ejemplo, la LV (26) *dormir como un tronco*, significa dormir profundamente, y acepta modificaciones paradigmáticas (dormir como un bendito, dormir como piedra), observando que su grado de fijación no es alto (no totalmente fija). Sin embargo, no es aparentemente posible definir a priori las variaciones realizables de una unidad fraseológica verbal. En el trabajo de [5] se hace énfasis que en las UF no hay presencia uniforme o ausencia de restricciones sintácticas, dado que no todas las unidades fraseológicas exhiben la misma versatilidad sintáctica. En [6] se postula que las secuencias de la misma estructura sintáctica no aceptan las mismas variaciones. Tomando en cuenta estas características, centramos nuestra

atención en la identificación de locuciones verbales, pues es una tarea que implica un reto importante en términos de la identificación y clasificación semántica. En [2] se realiza un estudio gramatical asociado a las locuciones verbales en español. Dicho estudio, “recoge y analiza una muestra de locuciones verbales del español distinguiéndolas claramente como tales de otras unidades fraseológicas que suelen confundirse con ellas”.

Las siguientes secciones presentan los experimentos computacionales realizados al intentar detectar automáticamente si existe una locución verbal en un texto plano, es decir, sin etiquetas o preprocesamiento alguno.

3. Marco metodológico

En cuanto a los avances actuales en la tarea de identificación automática de Locuciones Verbales (denotadas por LVS) para el español, se ha considerado el género periodístico mexicano y las LVS mexicanas (al denotar locuciones verbales mexicanas, se hace referencia a que son entendidas y utilizadas en México; esto no excluye que se empleen en otro país de habla española). Por lo tanto, en esta sección primero se describirán los recursos léxicos construidos para la tarea propuesta. El enfoque para la identificación de LVS está basado en técnicas supervisadas de aprendizaje automático, un área de la inteligencia artificial que concierne al estudio de sistemas computacionales que pueden llegar a aprender a partir de datos supervisado. Por lo tanto, también se incluye en esta sección una parte que describe los clasificadores empleados en los experimentos.

3.1. Recursos léxicos

Los métodos de aprendizaje automático supervisado asumen que se tienen datos anotados manualmente con los cuales se puede adquirir (aprender) conocimiento. En este caso, se necesitan corpora etiquetados manualmente por expertos indicando si cierto texto tiene o no una locución verbal. Por lo tanto, se ha construido un corpus supervisado para los experimentos propuestos en este artículo, seleccionando relatos periodísticos que contengan y no contengan locuciones verbales. Los relatos periodísticos están escritos en español mexicano y han sido recolectados durante el periodo del 2007 al 2013 de los periódicos mexicanos pertenecientes a la Organización Editorial Mexicana¹. Para este fin, primero se han extraído todas las locuciones verbales de un diccionario, versión papel, denominado “Diccionario de Mexicanismos”. Se ha seleccionado este diccionario porque sus elementos léxicos datan de la mitad del siglo XX y principios del siglo XXI (sincrónico), y compara lo que se dice en México con lo que se dice en otros países de habla española y sobre todo con el español de la Península Ibérica (contrastivo), indicando la realidad de uso sin excluir neologismos ni palabras de otras lenguas (descriptivo).

¹ Organización Editorial Mexicana, conocida como OEM, es la compañía de medios impresos más grande de México y una de las editoras de periódicos más grande de Latinoamérica.

En particular, se han recolectado 1,406 locuciones verbales de dicho diccionario que han sido almacenadas en una base de datos, considerando que éstas podrían ser posteriormente empleadas para la identificación de su uso regular en el género periodístico mexicano.

Con el uso de técnicas de recuperación de información se han encontrado 3,164 textos informativos, recolectados de la Organización Editorial Mexicana, que contienen al menos una ocurrencia de alguna de las unidades fraseológicas seleccionadas. Este proceso considera la ocurrencia original de la LV y/o alguna de sus variantes morfológicas. Por ejemplo, la LV (27) *darse por vencido* con sus variantes morfológicas (*darse por vencida*, *darse por vencidos*, *darse por vencidas*, *darnos por vencidos*, *darnos por vencidas*). Para este propósito, se han lematizado ambos conjuntos de datos, la LV y el texto de la noticia, con la finalidad de ser capaces de encontrar las variaciones de la LV en el texto objetivo.

Como consecuencia del conteo de la ocurrencia de las LVS mexicanas en el corpus recopilado, se ha construido un corpus etiquetado que además puede utilizarse como corpus de entrenamiento para los métodos de aprendizaje automático supervisado, con el objetivo de identificar si una noticia contiene o no una LV. Los contextos recopilados han sido manualmente anotados por 5 anotadores humanos, con un grado de acuerdo entre anotadores superior al 80 %. El proceso de la anotación manual realizado por estudiantes de computación dedicados a la lingüística computacional ha consistido en que un experto en lingüística explique que es una locución verbal y proporcione detalladamente sus características. Esta explicación fue basada en ejemplos, con el fin de diferenciar estas unidades fraseológicas de otras. Cada anotador humano fue encargado de clasificar manualmente, de acuerdo a las características de las LVS (descritas en la sección 2), cuando un texto plano contenía una LV (Clase 1) y cuando el texto no contenía una LV (Clase 2). El resultado de clasificación de las 3,164 noticias inicialmente recuperadas correspondió a 1,959 textos informativos para la Clase 1 y 1,205 para la Clase 2. Cada clase con 117,715 y 63,600 palabras, respectivamente.

En los experimentos realizados, todos los textos fueron representados mediante un vector de frecuencia de “n-gramas” de palabras, con $n = 1, 2$ y 3 . Frecuencias superiores a dos, para los “n-gramas” fueron consideradas en el vector de características. El corpus fue utilizado tanto para el entrenamiento como para las pruebas por medio de un mecanismo de validación cruzada a 10 pliegues (*10 fold-cross validation*).

3.2. Clasificadores empleados

Las técnicas de aprendizaje automático supervisado son capaces de aprender el proceso humano para identificar las unidades fraseológicas verbales con base en las características alimentadas por el corpus manualmente anotado. Con el fin de tener una perspectiva del tipo de clasificador que puede tratar mejor el problema de la identificación automática de locuciones verbales, se han seleccionado los siguientes cuatro algoritmos de aprendizaje (cada uno perteneciente a un tipo de clasificador diferente: Bayes, Lazy, Functions y Trees [17]):

1. Naïve Bayes: es un clasificador probabilístico basado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales.
2. K-Star: Este es el clasificador de los k vecinos más cercanos con una función de distancia generalizada.
3. SMO: Se trata de un algoritmo de optimización secuencial mínima para la clasificación de vectores de soporte.
4. J48: Es un algoritmo usado para generar un árbol de decisión.

4. Resultados experimentales y discusión

En esta sección se presenta la exactitud obtenida por cada clasificador cuando se trata de clasificar si existe o no una LV en un texto plano. Se ha usado la exactitud que una medida de evaluación estándar y una descripción completa de ésta puede ser consultada en [9].

En la Tabla 1, se muestran los porcentajes de las instancias que fueron clasificadas correcta e incorrectamente. Esta tabla, básicamente, muestra el promedio ponderado de los diferentes clasificadores empleados. La diferencia entre los valores de exactitud obtenidos por los clasificadores muestra que el rendimiento es bastante aceptable para ser una técnica computacional que se ejecuta de manera automática.

Tabla 1. Porcentaje de instancias clasificadas correcta versus incorrectamente.

Calsificador	Tipo	Instancias correctas (%)	Instancias incorrectas(%)
Naïve Bayes	Bayes	74.05	25.95
K-Star	Lazy	71.14	28.86
SMO	Functions	75.32	24.68
J48	Trees	76.74	23.26

Es importante remarcar que, en el estado del arte, las técnicas basadas en vectores de soporte obtienen regularmente los mejores resultados, lo que en este caso no resultó así. Es altamente probable que la técnica basada en árboles de decisión (J48) haya tomado ventaja de las estructuras de decisión para proponer un mejor rendimiento sobre el resto de las aproximaciones. En particular, el uso de múltiples tipos de “n-gramas” es un factor relevante en el proceso de clasificación que utiliza J48, puesto que asume la presencia de una gramática al considerar que una cierta palabra va antes o después de otra. Los valores obtenidos (alrededor del 76 %) indican que se han podido extraer algunas relaciones suficientemente generales como para generar hipótesis de comportamientos de los algoritmos de árboles J48 ante distintas situaciones, lo cual ha validado que se obtiene un porcentaje de efectividad similar para cada ejecución promediada mediante la aplicación del mecanismo de validación cruzada.

En cualquier caso, el experimento realizado demuestra la factibilidad de aplicar métodos computacionales para automatizar una tarea que puede incluso ser compleja para un ser humano, puesto que determinar sin ayuda de los seres humanos si una determinada frase es una locución verbal o no, definitivamente no es una tarea sencilla. Los resultados obtenidos son alentadores y se espera mejorarlos con la ayuda del enfoque lingüístico.

5. Conclusión

En los últimos años se ha incrementado el interés por la fraseología, una disciplina de la lingüística aplicada, debido a su importancia en diversas áreas del procesamiento de lenguaje natural como la traducción y el análisis de sentimientos, por citar algunos ejemplos. ésta pone en relieve diferentes aspectos tanto sociales como culturales de estas combinaciones de palabras. Las unidades básicas de análisis en la fraseología son las unidades fraseológicas o frasemas. Ciertas combinaciones de palabras no se pueden separar; pueden existir ciertas variantes, usos y modos de empleo de acuerdo a cada lengua, sin embargo, siguen estando unidas.

El análisis y estudio de las unidades fraseológicas, en específico locuciones verbales (que en este trabajo se han denominado locuciones verbales mexicanas), pone en relieve la complejidad y riqueza de la lengua española. Este artículo se ha enfocado principalmente a su estudio e identificación, expresándolo al presentar un conjunto de experimentos dedicados a su identificación automática en textos planos.

Para poder aplicar el enfoque basado en máquinas de aprendizaje supervisado, se ha recolectado una serie de unidades fraseológicas mexicanas usando el Diccionario de Mexicanismos. Estas unidades fraseológicas han servido como consultas de un sistema de recuperación de información que arrojó como respuesta un conjunto de textos (noticias) que contienen ocurrencias de dichas secuencias de palabras. Sin embargo, tal y como era esperado, no todas las secuencias hacen referencia a una locución verbal, y por tanto, fue necesario etiquetar manualmente todos y cada uno de los textos recuperados de Internet. El corpus supervisado que se obtuvo se utilizó como conjunto de entrada para cada uno de los clasificadores a evaluar.

Los siguientes, fueron los cuatro tipos de algoritmos de aprendizaje supervisado que se analizaron: Naïve Bayes, K-Star, SMO y J48. Se compararon estos clasificadores supervisados con el objetivo de determinar cuál de ellos presenta el mejor rendimiento. Se observa que la revisión 8 del árbol de decisión C4.5 fue la que obtuvo el mejor resultado para la tarea ejecutada en este artículo, obteniendo un valor de 76.74% de exactitud (instancias correctamente clasificadas). Este resultado es ligeramente mayor que el algoritmo SMO el cual obtuvo un valor de 75.32%, mientras que Naïve-Bayes y K-Star obtuvieron, 74.05% y 71.14%, respectivamente. Se está interesado en mejorar el rendimiento obtenido mediante el análisis de otras características que se pueden utilizar en el proceso de clasificación, por ejemplo, el grado de cohesión interna entre los componentes

de la locución verbal, o la posible baja correlación de dichos componentes con respecto a las palabras que se encuentran en el contexto de la locución verbal, sin embargo, este tema es considerado como trabajo futuro.

Referencias

1. Abeillé, A., Schabes, Y.: Parsing idioms in tree adjoining grammars. In: Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics, 1–9 (1989)
2. Bobes Soler, E.: Información gramatical asociada a las locuciones verbales del español. Tesis de Doctorado. Universidad de Barcelona, España (2016)
3. Carneado Moré, Z.: Algunas consideraciones sobre el caudal fraseológico del español hablado en Cuba. *Estudios de fraseología*, 7–38 (1985)
4. Carneado Moré, Z.: Algunas clasificaciones de la composición fraseológica de la lengua. *Anuario L/L*, 18, 34–45 (1987)
5. Casares, J. (Eds.): Introducción a la lexicología moderna. C.S.I.C. Madrid (1992 [1950])
6. Corpas Pastor, G. (Eds.): Manual de fraseología española. Madrid: Gredos (1996)
7. Coseriu, E.: Structure lexical et enseignement du vocabulaire. Dans: Actes du premier colloque international de linguistique appliquée, 175–217 (1966)
8. Dias, G.: Extraction automatique d’associations lexicales à partir de corpora. Tesis doctoral, New University of Lisbon, Portugal and LIFO University of Orleans, France (2002)
9. Fawcett, T.: Roc graphs: Notes and practical considerations for researchers. Technical report, HP Labs (2004)
10. García-Page Sánchez, M.: Introducción a la fraseología española. Estudio de las locuciones. Rubí (Barcelona), Antrhopos (2008)
11. González Rey, M.: La idiomática en las expresiones fraseológicas. En Wotjak, G. (ed.) *Estudios de fraseología y fraseografía del español actual*, Frankfurt-Madrid: Vervuert - Iberoamericana, pp. 57–73 (1998)
12. Gramley, S., Pätzold, K.M. (Eds.): *A Survey of Modern English*. Londres-Nueva York, Routledge (1992)
13. Gross, M.: Une classification des phrases “figée” du français. *Revue québécoise de linguistique* 11(2), 151–185 (1982)
14. Haddouch, B.: Las unidades fraseológicas en la enseñanza del español (caso del alumnado marroquí del instituto cervantes de tetuán). Tesis doctoral. Universidad Mohammed Vagdal. Facultad de letras y ciencias humanas, Departamento de estudios hispánicos-Rabat (1992)
15. Haensch, Wolg, G., Ettinger, L., Werner, S. (Eds.): *La lexicografía. De la lingüística teórica a la lexicografía práctica*. Gredos, Madrid (1982)
16. Lamiroy, B.: Le problème central du figement est le semi figement. *LINX*, pp. 135–153 (2005)
17. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA (2008)
18. Martins Baltar, M. (Eds.): *La locution entre langue et usages*. Fontenay- St. Cloud (1997)
19. Ortega, G., Gonzáles A.: En torno a la variación de las unidades fraseológicas, *Fraseología contrastiva: con ejemplos del alemán, español, francés e italiano*, pp. 91–109 (2005)

20. Priego Sánchez, B., Pinto, D., Mejri, S.: Metodología para la identificación de secuencias verbales fijas. *Research in Computing Science* 85, pp. 45–56 (2014)
21. Ruiz Gurillo, L. (Eds.): Un enfoque didáctico de la fraseología para extranjeros. Valencia: Quaderns de Valencia: Quaderns de Filología, Universitat de Valencia (2001)
22. Thun, H.: Probleme der phraseologie. Beihefte zur zeitschrift für romanische philologie 168 (1978)
23. Tristán Pérez, A.M.: Fuentes de las unidades fraseológicas. *Estudios de fraseología*, pp. 67–90 (1985)
24. Wotjak, G. (Eds.): *Estudios de fraseología y fraseografía actual*. Frankfurt: Vervuert/Iberoamericana (1985)
25. Zuloaga, A. (Eds.): *Introducción al estudio de las expresiones fijas*. Frankfurt: Verlag Peter Lang (1980)

Application of Natural Language Processing Techniques for Classification of Web Published News in Spanish

Yadira Hernandez-Cruz, Angel Chi-Poot, Gilberto Martinez-Luna

Instituto Politecnico Nacional, Centro de Investigacion en Computacion, CDMX, Mexico

Abstract. Web published news written in the Spanish language, were analyzed by using categories that are related to its content, such as: 'Culture', 'Sports' and 'Finances', or they are classified very generally as is the case of 'National' or 'International'. The large content of documents generated the need to provide the user with an analysis of such documents, particularly in circumstances where in search engines are involved. First of all, a pre-process was applied to allow the mining of texts, which includes the lemmatization, homologation of synonyms and representation of documents with a Boolean method. This pre-process also includes a dimensional reduction of the obtained matrix. Secondly, different classification methods were applied to compare their performance in order to find the one that best assigns the category to the news.

Keywords: natural language processing, machine learning, text mining, classification, dimensionality reduction.

1 Introduction

Today there is an accelerated growth of information published on the internet that can be recovered from multiple sources, topics and formats. In this work a categorization of news that was published on the Mexican web sites is analyzed by applying Natural Language Processing (NLP) techniques. This web sites publish thousands of informative articles, and it is an opportunity area to apply text mining. Given the large volume and diversity in the type of information published, it becomes necessary to create automated tools that facilitate the user navigation, retrieval, synthesis and analysis of documents. For that reason in this work the analysis of categories was carried out looking for a classification model with Bayesian Networks, Support Vector Machine and Multilayer Perceptron.

A particular characteristic of this type of information is the size of data to be analyzed. For example, after the application of NLP techniques, we obtain a big table with thousands of attributes. This matrix used to be a sparse matrix and consequently it is an excellent candidate for the application of dimensionality reduction techniques. In this paper the Correlation-Based Feature Selection (cfs_subset.eval) method was used for this purpose [3].

2 Related Work

This section explore the main researches related to news classification and the results obtained in them.

Different works have been carried out to categorize documents and news, such as the work presented in [5], where they use decision trees method, Naive Bayes, and Support Vector Machine, to classify news in Thai, obtaining a better F1 result equal to 95.42%, with the Support Vector Machine and modeling the documents under the information gain format. In contrast with this work, the results are similar but with a better time model creation because of the use of dimensional reduction.

The authors [10] propose a method of classifying web pages using a neural network, applying an analysis of main components, obtaining an average accuracy of up to 93.81% in sports news. On the other hand, this paper presents a comparison with other classification algorithms that showed different perspectives in the obtained results.

In another project, the researchers [9], categorized Indonesian news that talks about business, politics or sports, they used the technique of support vector machines, under the information gain model with the whom the achieved 98.057% accuracy. In comparison to this work, this paper uses not just 3 classification labels, it uses 6 of them.

3 Methodology

The goal of this section is to explore the elements used in this work, beginning with a general definition of natural language processing, classification and dimensionality reduction is provided. Also the specific classification algorithms used and (cfs.subset.eval) method are explained in a concrete way.

Natural Language processing is an empirical science that belongs to the area of humanities, relies heavily on different techniques of computer science and artificial intelligence, it is responsible for creating different models of spoken or written language to transform them into formal patterns that can be understood by computers, as well as being used in the construction and evaluation of hypotheses; once these models are obtained, they can be used to process the data sets by applying different techniques of machine learning that support the fulfillment of objectives such as automatic translation, elaboration of summaries, analysis of feelings, grouping, among others [11]. There are several techniques, but the most valued in this work are the following.

- Boolean representation of terms per document: It consists of a model of vector space in which text documents are transformed into a format that is suitable for analysis on a computer, so that it can be used to process documents, apply text mining algorithms and interpret the results generated. Within this model, each text document is represented by a vector where each characteristic is a term or a word, which has a value of 0 if the word is not found in the news, and 1 when it is found in the document [7].

- Elimination of stop words: The stop words are a set of words that provide little or no semantic meaning in the texts, they are generally the words that appear most frequently in a language and contain prepositions, pronouns, auxiliary verbs, etc. Eliminating stop words is a basic step in pre-processing to perform text mining, which, as the name suggests, consists of removing the stop words from the set of characteristics of the texts [1]. The catalog used contains 613 stop words in Spanish.
- Homologation of synonyms: Process of reduction of dimensions that is done grouping the words that have the same meaning. The synonyms dictionary contains the 3,820 most used words in Spanish according to the Royal Spanish Academy, each with an average of 3.7 synonyms, and a total of 14,103 synonyms of which 9,341 were used.
- Lemmatization: Change of each word by its simple form that must be an existing word, consists of eliminating the inflections, prefixes, suffixes, conjugations, and so on, trying to maintain the original meaning of the words.

Classification is a problem where it is received a set of training records that contain characteristics or attributes within the tuple and a classification label, from which we look for a classification model that allows us to identify the label that corresponds to the news on ones records that are arriving but with an unknown class [4].

Dimensionality reduction allows to reduce the dimensions used to search and describe classification models, this process is very useful when we work with a data set where the records (tuples) are very long (more than 20,000 attributes in this work), since it allows us to identify the subset of characteristics that have a high impact to determine which class each record belongs to, and thereby generate a compact and efficient classification model with low processing costs both in computational resources and in execution time [6].

Bayesian Networks consists of models in the form of directed graphs that include probabilistic information, where each of the nodes represents a random variable and the edges contain the relationships and indicate the probabilities of occurrence. These networks apply concepts of probability, graph theory, computation and statistics, when their edges are not directed they are known as Markov chains that are characterized by modeling independent events [2].

Support Vector Machine (SVM) is a set of supervised learning algorithms developed by Vladimir Vapnik and his team at AT & T laboratories. A SVM is a model that represents the sample points in space, separating the classes into 2 spaces as wide as possible by means of a separation hyperplane defined as the vector between the 2 points, of the 2 classes, closest to the one that is called vector support.

Multilayer Perceptron (MLP) is a supervised learning algorithm, which uses a function when it is training in a data set (See Equation 1). In this equation, n is the number of dimensions for the entry and m is the number of dimensions for the output:

$$f(x) : R_n \rightarrow R_m. \quad (1)$$

Correlation-Based Feature Selection: This algorithm uses the correlation or co-variance of the attributes. Basically computes the correlation matrix between all the attributes and, based on that, selects those attributes that are strongly correlated with the classification label but weakly correlated between themselves [3]. This algorithm is useful for sparse matrices, but it takes a high time and consumes a lot of memory.



Fig. 1. News categorized in 'National' that should be classified as 'Politics'. The translation from Spanish to English is described below, nacional: national; Elecciones sin, encanto ni propuesta: Elections without charm or proposal; Un análisis revela conflictos partidistas como una constante: An analysis reveals a constant partisan conflict.

4 Dataset

In this work the database analyzed was Noti-Explorer [8] that corresponds to a set of news in Spanish published in free sites on the Internet, which can be seen to have categories that may be clearly related to its content, such as: 'Culture', 'Sports' and 'Finances', or they are classified very generally as is the case of 'National' or 'International'. This database contains news published on Mexican websites, with documents written in Spanish.

Particularly, this database is too large (258,726 news collected until May 13, 2019) and continues to grow with the daily collection of news, and consequently only the news published from January 1, 2019 to May 13, 2019 were analyzed. In order to train the classification models, news tagged with the following categories were used: 'Science and technology', 'Culture', 'Entertainment', 'Finances', 'Health' or 'Security'. Disregarding those with very general categories such as: 'National', 'International', or the name of any Mexican state. Table 1 shows the number of documents by category. Images 1 and 2 show examples of news that should belong to the categories of 'Politics' and 'Security' respectively, but were labeled as 'National'.

After the pre-processing, the news have an number of words average of 127, with a maximum of 886 and a minimum of 14. Figure 3 shows a graph with the distribution of the news size.

Table 1. Number of documents by category.

CATEGORY	Number of documents
Aguascalientes	3.185
Baja California	655
Ciencia y Tecnología	646
Coahuila	2.119
Cultura	122
Deportes	6
Durango	630
Entretenimiento	2.311
Finanzas	647
Internacional	2.028
Nacional	3.084
Nayarit	49
Salud	67
Seguridad	2.248
Sonora	1.120
Tamaulipas	41

5 Results

This work was done in two steps. First, pre-processing procedures were implemented having a huge attention to the natural language techniques. Second, the results of the classification algorithms are showed.

In the pre-processing step, natural language processing (NLP) and dimensionality reduction techniques were applied. That was done in order to reach a better structure of the data for the use of the classification algorithms. The following Natural Language Processing techniques were used:

- Elimination of stop words
- Lemmatize
- Homologation of synonyms
- Boolean representation of terms per document

Additionally, after the implementation of the procedures above, a sparse matrix was obtained where the rows are news and columns words included in



Fig. 2. News of 'Security' categorized as 'National'. The translation from Spanish to English is described below, nacional: national; Encuentran asesinados a policías plagiados en Jalisco: They find murdered plagiarized police in Jalisco.

the new (attributes). At the end of this matrix, the classification label was added. Finally, a matrix of 21,087 columns and 6,040 rows were created.

At the end of the step, dimension reduction was applied with the `cfs_subset_eval` algorithm to the matrix obtained. This process found 80 of 21,087 attributes are strongly correlated to the classification label but weakly correlated between themselves. This subset of data is appropriate to train classification algorithms that work better with few attributes such as Bayesian networks, as well as to run tests with the other algorithms.

After the pre-processing step, the matrices obtained were given as an input to the classification algorithms. Each algorithm had particular performances which are described in the following paragraphs.

Support Vector Machine trains the model with 80% of documents and perform tests on the remaining 20%. The total generated dimensions was 21,087. The average precision obtained was 83.42%. The precision detail by category is shown in Table 2.

Table 2. Precision achieved with Support Vector Machine by category.

CATEGORY	Precision
Ciencia y Tecnología	89,75%
Entretenimiento	60,57%
Finanzas	90,00%
Salud	98,92%
Seguridad	63,55%
Cultura	97,76%

The Multilayer Perceptron achieved the best performance with a precision of 94.4628% over the whole data set taking an approximate time of 15 minutes to

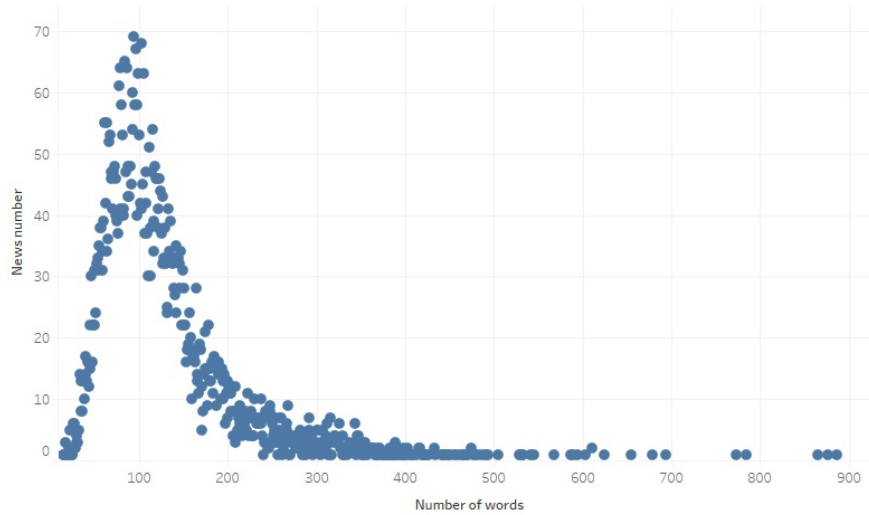


Fig. 3. Word count distribution of the news.

train the model, classify the test samples and calculate the accuracy. 80% of the data was used for training and 20% for tests, choosing them randomly. Another test was generated by adding a reduction in dimensions in the pre-processing, reducing the processing time to approximately 5 seconds. The test results carried out are in Table 3.

Table 3. Results generated with the multilayer perceptron model.

Number of Documents	Number of Dimensions	Precision
2.195	21,087	87,01%
6.040	80	81,98%
6.040	21,087	94,46%

The Bayesian network obtained 82.46% accuracy on the 6,040 documents and the 80 dimensions resulting from the reduction of characteristics. The model trained with 10 fold cross validation. In this case, the 21,087 matrix could not be analyzed because it required too many RAM memory (it reached more than 28 Gb of RAM).

6 Conclusion

Summarizing, it was possible to label the news documents up to 94% accuracy. The natural language processing techniques applied in the pre-processing step

allowed a better scenario for the classification and reduction algorithms. The classification by category was applied to the news that are with very general categories to give the user an idea of the content of the documents and reduce the search space as well as providing more information to generate an analysis.

The classifier that showed the best performance was the multilayer perceptron. Particularly, reduction of dimensions help us to generate classification models such as Bayesian Networks, which are complex to calculate with many attributes, while reducing processing time in the execution of learning algorithms. Reducing the dimensions in a general manner reduces the accuracy of the classification, but improves the time required for model creation.

References

1. Amarasinghe, K., Manic, M., Hruska, R.: Optimal stop word selection for text mining in critical infrastructure domain. In: 2015 Resilience Week (RWS). pp. 1–6. IEEE (2015)
2. Ben-Gal, I.: Bayesian networks. *Encyclopedia of statistics in quality and reliability* 1 (2008)
3. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. Ph.D. thesis (1999)
4. Harrington, P.: *Machine learning in action*. Manning Publications Co. (2012)
5. Haruechaiyasak, C., Jitkrittum, W., Sangkeetrakarn, C., Damrongrat, C.: Implementing news article category browsing based on text categorization technique. In: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. vol. 3, pp. 143–146. IEEE (2008)
6. Hernández, J.A.: Métodos de reducción de dimensionalidad: Análisis comparativo de los métodos apc, acpp y acpk. *Uniciencia* 30(1), 115–122 (2016)
7. Onoda, T., Murata, H., Yamada, S.: Comparison of performance for svm based relevance feedback document retrieval in several vector space models. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 03. pp. 169–172. IEEE Computer Society (2008)
8. Ortega Castellanos, P.R.: Sistema de análisis visual para la exploración de grandes corpus periódicos utilizando modelación de tópicos y entidades nombradas (2017)
9. Rizaldy, A., Santoso, H.A.: Performance improvement of support vector machine (svm) with information gain on categorization of indonesian news documents. In: 2017 International Seminar on Application for Technology of Information and Communication (iSemantic). pp. 227–232. IEEE (2017)
10. Selamat, A., Yanagimoto, H., Omatu, S.: Web news classification using neural networks based on pca. In: Proceedings of the 41st SICE Annual Conference. SICE 2002. vol. 4, pp. 2389–2394. IEEE (2002)
11. Sidorov, G.: *Construcción no lineal de n-gramas en la lingüística computacional*. Mexico DF: Sociedad Mexicana de Inteligencia Artificial (2013)

Comparative Analysis of Interest Point Detectors Algorithms on Robotic Operative System

Francisco Javier Valdepeña Rivera¹, Dante Mújica Vargas², Miguel Ángel Ruíz¹

¹ Universidad Politécnica del Estado de Morelos
Jiutepec, Morelos, Mexico

² CENIDET-TecNM, Depto. Ciencias computacionales
Cuernavaca, Morelos, Mexico

{vrfo150252, mruiz}@upemor.edu.mx, dantemv@cenidet.edu.mx

Abstract. Applications of robotic vision have had great advances within the artificial intelligence through the processing of images, as well as the automated systems (robots). A comparative analysis of some interest point detector algorithms will be performed, the next analysis will be about a robotic operating system called ROS, by means of a 2D object detector system. For this purpose, a physical architecture will be carried out to carry out the experimentation within a controlled work environment, in order to demonstrate which algorithm will work best in the future for the development of object recognition systems, implementing this system on Robots.

Keywords: detector, interest points, ROS, artificial vision.

1 Introduction

The artificial vision in these days has been realized through systems that detect characteristics of the objects and classify them by means of neural networks which decide what type of object is, by means of a comparison of weights or learning units which are updated every that an iteration within algorithm.

Deep learning is based on a set of algorithms for machine learning that attempts to model high-level abstractions in data using architectures composed of multiple non-linear transformations [1].

But there is also the way to segment artificial vision in various stages which are essential to divide the number of processes into more concise processes, such as those shown in Figure 1. This is to perform optimal recognition in robotic systems, contrary to only have neural networks for recognition what it is to load a neural network with points or characteristics that are a stage of artificial vision can achieve, that the recognition is more precise and with a minimum of variety to the conditions in which find the object.

In this way it is necessary to take into account which are those algorithms that can extract the most important characteristics to be able to make a good detection, so after having said information the recognition is achieved instantaneously with a neural network or applying the concept of deep learning.

It is proposed to analyze and compare object detection algorithms under rotation, translation and light conditions to conclude which of them is the best for the task and

thus be able to have the analysis data and say which one to use in the future for object recognition.

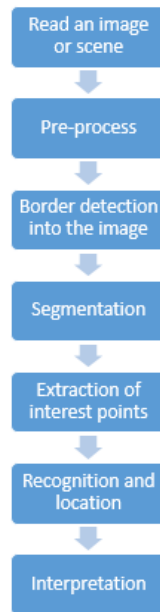


Fig. 1. Stages of artificial vision.

2 Theoretical Framework

2.1 Interest Point Detector and Feature Descriptor

Many algorithms that focus on robotic vision are based on the location of the points of interest or key points in each image, as well as calculating a description of the characteristic of the pixel region surrounding the point of interest. This contrasts with methods such as correlation, where a larger rectangular pattern is passed over an image within the pixel intervals and the correlation is measured at each location.

The point of interest is the anchor point, as it often provides the attributes of scale, rotation and illumination invariance for the descriptor.

A point of interest can be composed of several types of corners, edges and maxima, as shown in Figure 2.

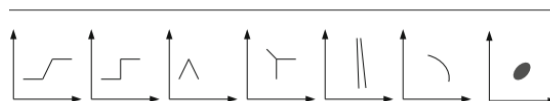


Fig. 2. Interests point [2].

2.2 Interest Points Detectors

FAST

It is a corner detector that is characterized by producing very stable relevant points. This method belongs to the category AST (Accelerated Segment Test), It is the most efficient computationally corners detector that exists so far. This algorithm works through circular windows centered on each of the points of the image. For a point to be detected as a corner there must be a minimum number of points within its window that are a threshold amount more clear than the center or another threshold amount darker than the center [4].

The segment test criterion operates considering a circle of 16 pixels' candidate for the return p. The original detector classifies p as a corner for automatic detection of high-speed corners 5 if there is a set of n contiguous pixels in the circle that are all brighter than the intensity of the candidate pixel I_p plus a threshold t, or everything else dark than $I_p - t$, as illustrated in Figure 3.

n It was chosen to be twelve because it supports a high-speed test that can be used to exclude a large number of non-corners: the test examines only the four pixels in 1, 5, 9 and 13 (the four directions of the compass). This detector itself exhibits high performance, but there are several weaknesses:

- Do not reject so many candidates for $n < 12$.
- The choice of pixels is not optimal because their efficiency depends on the order of the questions and the distribution of the appearances of the corners.
- The results of the high-speed tests are discarded.
- Multiple features adjacent to each other are detected.

SIFT

The SIFT algorithm is an artificial vision algorithm originally published by David Lowe in 1999, which is responsible for extracting distinctive features from grayscale images. It is used primarily for reconnaissance tasks. The selected points are invariant to scale, rotation and partially invariant to changes of point of view and illumination. The extraction algorithm is based on the location of the maximum and minimum of a Gaussian difference applied in the space of scales. This feature extraction allows the algorithm to be executed in a hierarchical way so that the characteristics are calculated from a pyramid of images with sampling between each level [2].

Detection of extremes in space-scale

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (1)$$

Here:

$$L(x, y, \sigma) = I(x, y) * G(x, y, \sigma), \quad (2)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}. \quad (3)$$

Exact location key point

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x. \quad (4)$$

Here D and its derivatives are always evaluated at the point in question and $x = (x, y, \sigma)^T$ is the position relative to it.

Deriving the previous approximation and equaling it to zero, we obtain:

$$\underline{x} = -\left(\frac{\partial^2 D^{-1}}{\partial x^2} * \frac{\partial D}{\partial x}\right). \quad (5)$$

The previous formula is replaced with the first Taylor series of grade 2, to obtain the local maximum value:

$$D(\underline{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \underline{x}, \quad (6)$$

$$Traza(H) = \frac{\partial^2 D}{\partial x^2} + \frac{\partial^2 D}{\partial y^2} = \alpha + \beta, \quad (7)$$

$$Det(H) = \frac{\partial^2 D}{\partial x^2} * \frac{\partial^2 D}{\partial y^2} = \alpha * \beta. \quad (8)$$

Let $\alpha = r.\beta$, the condition is reduced to:

$$\frac{Traza(H)^2}{Det(H)} < \frac{(r + 1)^2}{r}. \quad (9)$$

See that as the relation r between both eigenvalues increases, so does the relation between the square of the trace of the Hessian matrix and its determinant.

Orientation assignment

By assigning an orientation to each point of the image based on local characteristics of the image, the key points can be described relative to these orientations and in this way achieve invariant characteristics to the rotations.

Where to calculate the gradient m (x, y) is:

$$m(x, y) = \sqrt{(\Delta L_x)^2 + (\Delta L_y)^2}. \quad (10)$$

Here m (x, y) equals to:

$$m(x, y) = \sqrt{[L(x + 1, y) - L(x - 1, y)]^2 + [L(x, y + 1) - L(x, y - 1)]^2}. \quad (11)$$

Where to calculate the phase θ (x, y), we use:

$$\theta(x, y) = \left(\frac{\Delta L_y}{\Delta L_x} \right). \quad (12)$$

Here $\theta(x, y)$ equals:

$$\theta(x, y) = \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right). \quad (13)$$

SURF

The SURF algorithm consists of the following phases:

- Detection of interest point.
- Assignment of orientation to interest point.
- Calculation of descriptors associated with interest point [2].

Detection of Interest point

$$H(p, \sigma) = [L_{xx}(p, \sigma) \ L_{xy}(p, \sigma) \ L_{xy}(p, \sigma) \ L_{yy}(p, \sigma)]. \quad (14)$$

Where $L_{xx}(x, \sigma)$ is the second-order convolution of the Gaussian, $\partial^2/(\partial x^2) g(\sigma)$ with the image I at point x , and similarly for $L_{xy}(p, \sigma)$ $L_{yy}(p, \sigma)$.

The approximations of the partial derivatives are denoted as D_{xx} , D_{xy} and D_{yy} , and the determinant is calculated as follows:

$$\det \det (H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2. \quad (15)$$

Assignment of orientation to interest point

The first step to grant the mentioned orientation consists of the calculation of the answer of Haar in both directions x and y .

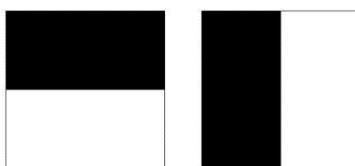


Fig. 3. Calculation of the answer Haar. Black: -1 White: +1 [2].

After having made all these calculations, integral images are used again to proceed to the filtering through the Haar masks and thus obtain the answers in both directions. Only 6 operations are necessary to obtain the answer in the x and y directions. Once the wavy responses have been calculated, they are weighted by a Gaussian value of $\sigma = 2.5s$ centered on the point of interest.

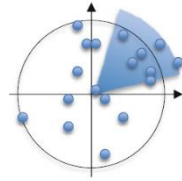


Fig. 4. The sliding sector window used in SURF to calculate the dominant orientation of HAAR [2].

Calculation of descriptors associated with points of interest

In summary, each of the sub regions is represented by a vector of components:

$$v = \left(\sum \partial x, \sum \partial y, \sum |\partial x|, \sum |\partial y| \right). \quad (16)$$

ORB

ORB is based in part on the BRIEF descriptor, thus the name Oriented Brief, since ORB adds orientation to the BRIEF method and provides other improvements as well.

In BRIEF, as shown in Fig. 5, the sample points are specified in a random distribution pattern based on a Gaussian distribution about the center point within the 31 x 31 patch region; the chosen number of sample points is 256. Selected sample point-pairs are compared to each other to form the binary descriptor vector. The value of each point is calculated via an integral image method to smooth a 5 x 5 region into the point value.

The ORB pattern shown in Fig. 5 is based on choosing point-pairs that have high statistical variance within a bounding 31 x 31 image patch, where the smaller 5 x 5 gray image patch regions are centered at the chosen interest points. Then each 5 x 5 region is smoothed using an integral image method to yield a single value for the point [2].

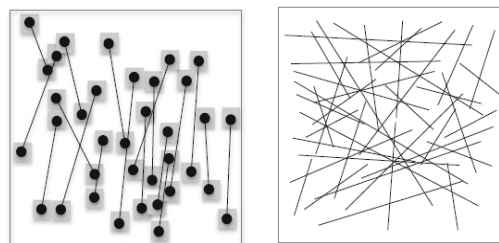


Fig. 5. (Left) An ORB style pattern at greatly reduced point pair count resolution, using < 32 points instead of the full 256 points (Right). A BRIEF style pattern using randomized point-pairs [2].

It is a collection of tools, libraries and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

The computation graph level

In the Figure 6 shows the computation peer-to-peer of the ROS process.

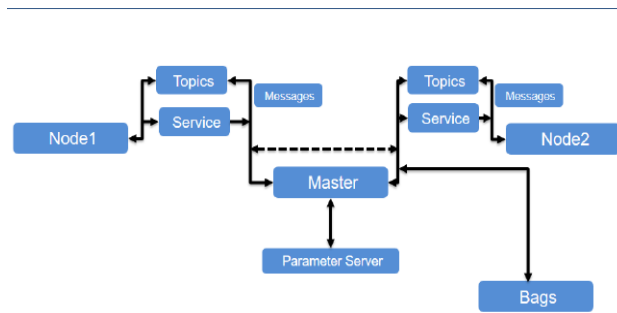


Fig. 6. Computation graph levels [3].

3 Methodology

This section explains how the system is working. The interaction begins when the user enters the system the find_object_2d, which in turn has the folder configuration that ROS manages, that is, the system itself controls the hardware of both the webcam and the computer that functions as the server of the own system, that is why there is a direct and parallel interaction with the system and the devices, working together it receives the images taken from the test object in order to have the data that finally arrive at the find_object_2d system and are processed with the algorithms SIFT, SURF, FAST, ORB which will detect if the camera captures the object or not, the architecture of the system is shown in figure 6.

3.1 Functionality of the System the Find_object_2d

The find_object_2d system works from the communication of the webcam node which is in the variable /usb_cam/image_raw, it is necessary to have installed the webcam driver compatible with ROS, if you do not have the driver, you must use the following command [3]:

```
$ sudo apt-get install v4l-utils
```

Once having the webcam driver, the next thing is to start the communication of nodes within ROS, as shown in the figure 7:

```
$ roscore
```

Plug your USB camera into your PC, and launch the ROS usb_cam driver:

```
$ roslaunch usb_cam usb_cam-test.launch
```

Then the next step is to run the object detector node. The following command will start the object detector node:

```
$ rosrunc find_object_2d find_object_2d image: =/usb_cam/image_raw
```

Finally, after inserting the above commands without any problem, the detector system will show the following window as shown in figure 7.

The window is divided into 3 panels.

In the first panel the images of the events captured by the camera are inserted and those images will be compared with the image in panel 2 if they are in both panels the same object the algorithm must detect if there is a match object.

In the third panel are the basic configurations of the detector system. In that panel the different algorithms will be inserted, to perform the comparative analysis of interest point detectors, the option to perform the insertion of the algorithm is Feature 2D.

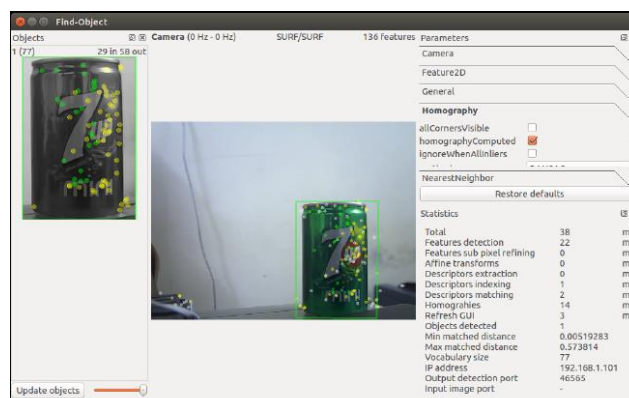


Fig. 7. Main window of the The_find_object 2d system [3].

3.2 Considerations

To perform the comparative analysis of the different algorithms, the following hardware and software points were taken into account.

Hardware

- With a computer with the following minimum characteristics.
 - o Intel processor with a speed of 2 to 3 GHz.
 - o RAM memory with a minimum capacity of 2 GB, recommended 4 GB.
 - o Hard disk with a minimum storage capacity of 500 GB.
 - o Have common input / output peripherals (mouse, monitor, keyboard).
- Count a digital camera with a minimum resolution of 640 x 480 pixels with USB input.
- (Optional) Infrared sensor with VGA camera (Kinect Xbox). Said device will be implemented to calculate distances between the object and the image receiver if the system requires it.

Software

- Linux distribution operating system recommended an Ubuntu LTS version 16.04.
- Activate updates of Ubuntu LTS 16.04, to update devices drivers
- Install ROS Kinect kame.

4 Obtained Results

Table 1. Characteristics of a controlled environment in the experimentation test.

Change	1	2	3	4	5
Illumination	75%	75%	75%	40%	40%
Angle of rotation	0°	0°	45°	0°	0°
Number of objects	1	3	1	1	3

Table 2. Analysis of points by pixels in the SIFT and SURF algorithms.

Objects	SIFT			SURF		
	Average	max	min	average	max	min
One object	95.065%	98.571%	81.904%	87.582%	97.619%	81.008%
Three objects	90.906%	98.571%	64.577%	91.482%	97.058%	85.401%
A rotated object	87.197%	97.744%	67.057%	83.66%	95.555%	60.191%

In the table 2 show data of points by pixel that means the interest points that every algorithm, that is to say the quotient of the interest points on all points detected.

In section of the algorithm SURF where the average is 91.482% shows an increment with regard to the algorithm SIFT is for this reason that the algorithm SURF detects better than algorithm SIFT. Every point by pixel is a right matching that means interest point and finally detected objects.

A characteristic of the algorithm SIFT is the quantity of points by pixel to detect objects and it's not the more important to detect a matching or interest point.

The implementation in the algorithm SURF about of points eliminated with a condition of maximum between scales neighbors change the maximum interpolating between scale and the octave position.

Table 3. Analysis of points by pixels in the FAST and ORB algorithms.

Objects	FAST			ORB		
	average	max	min	average	max	min
One object	2.352%	3.253%	0.216%	75.370%	85.106%	59.788%
Three objects	1.177%	3.770%	0.0714%	73.635%	82.954%	57.971%
A rotated object	0.604%	1.449%	0.0907%	74.378%	82.954%	61.111%

In the table 3 show data of points by pixel of FAST and ORB algorithm to detect objects in this proof we observe interesting results although the FAST algorithm detects a big quantity of data or points by pixel, the percent to detect interest points is low because is more sensible by the conditions environment we lost the umbral and this situation changes the result of matching.

Another example was ORB algorithm because if you change the rotation or illumination on the object, you have a partial matching that is to say in the next 3 seconds approximately the algorithm loses interest points to detect the object.

However, the answer of ORB algorithm is faster than FAST algorithm. But FAST algorithm has a better detection of points by pixel than ORB and it's less sensible on rotation of objects.

5 Conclusions

In conclusion, every algorithm has characteristics that improve the detection of objects through of interest points but some algorithms weren't showing results waited in the theory.

The SIFT and SURF were constant in their results but the SURF algorithm have a better performance than SIFT, with less points by pixel was able to detect all objects in a short answer of processing and all environments of proof. But the best conclusion about of SURF algorithm was the invariability of rotation, illumination and kind of object is for this reason that SURF has an important position in the artificial vision as detector of interest points for the recognition of objects in uncontrolled environments.

References

1. Alvear, V., Farinango, H., Navarrete, I., Rosero-Montalvo, P., Noguera, J., Cuzme, F., Peluffó, D.P.: Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. INCISCOS 2016, pp. 113-116 (2016)
2. Krig, S.: Computer vision metrics. Switzerland: Springer International Publishing (2016)
3. Joseph, L.: Projects, ROS Robotics. Birmingham, UK: Packt Publishing Ltd. (2017)
4. Sucar, E.: Visión computacional. México: Instituto Nacional de Astrofísica, Óptica y Electrónica (2017)

Convolutional Model with Classification through Izhikevich Neuron

Antonio Luna-Álvarez, Dante Mújica-Vargas, Manuel Mejía-Lavalle

Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, Mexico
{jesus.luna18ce,dantemv,mlavalle}@cenidet.edu.mx

Abstract. This paper presents the fusion of two paradigms of neural networks: the Convolutional Neural Networks from deep learning and the third generation Izhikevich neuron. This fusion has the purpose to replace the multilayer perceptron layers, that usually represent a computational cost and a large training time, for a paradigm created to classify with a single neuron. The experimentation is carried out in the classification domain, predicting the directions of rotation in a simulator of self-driven vehicles. The experiments show similar results to the multilayer perceptron model in the evaluation metrics but the training time is reduced.

Keywords: Izhikevich neuron, convolutional neural network, deep learning, steering classification.

1 Introduction

Deep Neural Networks is one of the most used techniques nowadays for the object detection and classification in digital images, because they do not require specific parameters for each type of image, they adapt to the image in a automatic. However, one of the great disadvantages of this paradigm is the slow training process, which even with hardware of great processing capacity, requires hours or days of processing. Another limitation is the need for large databases to train and be able to perform the task successfully.

To reduce the training time of a Deep Neural Network, several solutions have been proposed. One way to reduce time is to reduce the training data, a related paper is [4] in which it is proposed to make a selection of representative data with heuristic values granted by human experts, this method decrease the training time without affecting the performance of the neural network. There are also more intelligent proposals in which techniques accelerate the learning of the network. For example in [18] is proposed to use an optimization technique based on Particle Swarm for the backpropagation, decreasing training epochs. Another way to accelerate learning is by intelligently tuning of the training parameters using sophisticated techniques like the shown in [17], which presents an optimization of the learning rate, and likewise to the previous proposal, decreasing the training epochs.

A less sophisticated but effective alternative to reduce training time is to increase the hardware used for this task, for example in [7] a framework is created to distribute the processing in a cluster of servers. And on the other hand, the use of deep neural networks in embedded systems has been very difficult due to the need for powerful hardware, which has forced to improve the devices in which these systems are used, as in the work of [10].

Although the proposals of the state-of-the-art present acceptable results there are some complications. In the first place, using only representative data might cause the network not to be able to classify correctly those objects that are not very recurrent in the training data or that resemble the most common ones. On the other hand, although intelligent methods for learning acceleration are the ideal technique to reduce the training epochs, it implies a second processing and as is well known, the convergence of the Particle Swarm algorithm requires a long time due the complexity of $O(n \cdot p + Cof \cdot p)$ where n is the dimension of the problem p is the population size and Cof is the cost of the objective function. For this reason, although they reduce epochs, training time can be increased. Finally, enhancing hardware is not the best option when you do not have the necessary resources.

A solution proposal more to the problem is the implementation of a updated neuronal paradigm. In this document is proposed the fusion of a Convolutional Neural Network for the extraction of information from digital images and the Izhikevich Neuron [8] used as a control model. The combination of the deep learning paradigm and the third-generation neuron allows a reduction of the training time without using algorithms or additional methods to the training of this neuronal model. The fusion of these paradigms allows to reduce the number of neurons in the classification layers (multilayer perceptron) thus causing a reduction of the processing without affecting the performance in the classification.

The following sections of the document sections are organized as follows. The Section 2 describes the theoretical concepts necessary to understand the functioning of convolutional layers in a deep neural network. The description of the third generation Izhikevich neuron is shown in the Section 3, as well as the fusion of this with the convolutional layers. The data used for the experimentation, the methods of evaluation and the configurations of the experiments are presented in the Section 4. Finally the results obtained and the comparison is made in the Section 5 and the conclusions in the Section 6.

2 Background

2.1 Convolutional Network

This type of neural network has great advantages of automation in the task of extracting image information. In this paradigm, no known image processing filter is applied, but through the training, it learns the image processing filters. In general, the first convolutional layer learns to detect edges, while the second can learn to detect more complex shapes that can be formed by combining

different edges, such as circles and rectangles. The third layer and beyond learn much more complicated features based on the characteristics generated in the previous layer [14].

The Convolutional Neural Networks (CNN) are based on the convolution of signals in two dimensions and on the detection of filter-based characteristics (called kernel) that they learn through training. To extract the information, the kernel performs the 2D convolution operation, expressed in the Equation 1, to the input image. Where N and M are the dimensions of the image, n_1 and n_2 represent the column and row indices of the pixel being processed, k_1 and k_2 the kernel indexes [12]:

$$y(n_1, n_2) = \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{M-1} x(k_1, k_2)h(n_1 - k_1, n_2 - k_2) \quad (1)$$

$$0 \leq n_1 \leq N - 1, 0 \leq n_2 \leq M - 1.$$

To train the convolutional layers, the backpropagation of the error technique is used through the convolutional layers. This is very similar to the backpropagation for a multilayer perceptron network, the difference is that the weight connections are scattered, since the different input areas share the same weights to create a map of output characteristics. In general, the feature map is obtained through the function expressed in (2). Where w represent the kernel weights and a a first feature map obtained. On the other hand, the kernel weights are adjusted by calculating the descendant gradient which is expressed in (3). Where L represents the error function:

$$S_{ij} = \sum_{n=1}^2 \sum_{m=1}^2 w_{(3-m)(3-n)} \cdot a_{(i-1+m)(j-1+m)}, \quad (2)$$

$$\frac{\partial L}{\partial w_{ij}} = \sum_{j=1}^2 \sum_{j=1}^2 \frac{\partial L}{S_{ij}} \frac{\partial S_{ij}}{\partial w_{ij}}. \quad (3)$$

In the generality of the Convolutional Neural Networks, there are layers of dimensionality reduction; Max Pooling layers are given by a discriminant function where the maximum value is selected in each region of the image by moving a kernel through it, thus generating a new feature map of smaller dimensions to the original. The function of this filter is given in (4), where n is the kernel dimension, N is the dimension of the image and $u(\cdot)$ is the window function:

$$a_j = \max_{N \times N} (a_i^{n \times n} u(n, n)). \quad (4)$$

On the other hand, there exists layers of Average Pooling, which follows the same principle as the previous one, with the difference that in this, the kernel

take an average of the region according to (5) [16]:

$$a_j = \frac{\sum_{N \times N} a_i^{n \times n}}{N^2}. \quad (5)$$

2.2 Izhikevich Neuron

This model emulates the pulsation behavior of a certain type of neurons known as cortical. The model combines the biological plausibility of Hodgkin-Huxley-type dynamics and the computational efficiency of integration and firing neurons (spiking networks) [9]. This neuron has the ability to perform non-linear classification unlike its predecessor, the simple perceptron. The model in question has been evaluated in multiclass [15] and non-linear applications [13].

Algorithmically this neuron depends on a series of parameters, which are distributed in three equations [6]; (6) represents the energy gain of the neuron, in (7) recovers the energy through time and finally (8) emulates the pulse of the neuron to reach a certain threshold and the restart of the neuron gains [5]:

$$C \frac{dv}{dt} = k(v - v_r)(v - v_t) - u + I, \quad (6)$$

$$\frac{du}{dt} = a(b(v - v_r) - u), \quad (7)$$

$$v \geq v_{peak} \Rightarrow \begin{cases} v \leftarrow c, \\ u \leftarrow u + d. \end{cases} \quad (8)$$

The weight adjustment is one of the contributions of [5] to Izhikevich's original model. This adjustment is given by calculating the difference in pulses obtained and expected, multiplied by the inner product of the synaptic weights w and the inputs x , scaled by the learning rate α . However, this method works correctly for the binary classification, but for a classification of multiple classes it is necessary to make an extension of this function using the medium, as shown in (9):

$$\Delta w = \alpha \cdot \sum_{i=0}^n \frac{1}{n} (y - y')^2 \cdot (w \times x). \quad (9)$$

In general, an Izhikevich neuron is trained through N epochs, for each the calculation of synaptic currents I is made based on the inner product of the weights and the inputs plus the bias, this synaptic current serves input for the calculation of the membrane potential of the v neuron that accumulates through 1000 iterations, achieving y pulsations. The algorithm of a single epoch is defined as shown in Algorithm 1.

Algorithm 1 Izhikevich neuron algorithm.

Input: $x \in \mathbb{R}^n, y' \in \mathbb{Z}$
Output: $y \in \mathbb{Z}$
 $I \leftarrow [(w \times x) + 1]^2 + \theta$
 $u \leftarrow 0$
 $v \leftarrow v_r$
for $i \leftarrow 0$ **to** $i = 1000$ **do**
 $v \leftarrow \frac{v + [k(v - v_r) - (v - v_r) - u + I]}{C}$
 $u \leftarrow d[b(v - v_r) - u]$
if $v \geq v_{peak}$ **then**
 $v \leftarrow c$
 $u \leftarrow u + d$
 $y \leftarrow y + 1$
end if
end for
 $E(w) \leftarrow \frac{1}{n} \sum_{i=0}^n (y_i - y'_i)^2$
 $\Delta w \leftarrow \alpha \cdot E(w) \cdot (w \times x)$
return y

Here $x \in \mathbb{R}^n$ represents the input vector, $y \in \mathbb{Z}$ the number of pulses emitted by the neuron and $y' \in \mathbb{Z}$ the expected pulses. $w \in \mathbb{R}^n$ represents the weights that match the inputs. The variable $v \in \mathbb{R}$ represents the membrane potential of the neuron and $u \in \mathbb{R}$ represents a variable of membrane recovery, which explains the activation of ionic currents $k \in \mathbb{R}$ and the inactivation of ionic currents, and provides negative feedback to v .

3 Convolutional Network and the Izhikevich Neuron Fusion

As mentioned in the Section 1, the use of another type of neuron than the perceptron in the classification layer reduce the training time. To extract information from digital images, it is still necessary to use a Convolutional Neural Network, however it is possible to replace multilayer perceptron layers, in this case by a spiking Izhikevich neuron.

In general, the fusion occurs when the numerical information obtained in the process carried out by the CNN is extracted. This numerical information is introduced as a synaptic vector to the Izhikevich neuron, similar to how it is done with the simple perceptron. The neuron performs the process mentioned in Algorithm 1 during training N epochs to adjust the weights w . Unlike the perceptron, the Izhikevich neuron does not return predictions given by the function $\sigma(\cdot)$ where $y' \in \mathbb{R} \rightarrow [0, 1]$ or by the function $\tanh(\cdot)$ where $y' \in \mathbb{R} \rightarrow [-1, 1]$, but this neuron returns *pulses* which comprise $y' \in \mathbb{Z} \rightarrow [0, \infty]$. So in the implementation was necessary a transformation of the labels of each class.

The implementation was handled as two dependent modules, a function was implemented that communicates both modules. From the CNN, extract the vector of characteristics obtained from the image and in the opposite direction normalize the cost function (Mean Squared Error mentioned in the Section 4) obtained from the Izhikevich neuron and introduced into the CNN for the backpropagation. The fusion of these modules is seen in Figure 1.

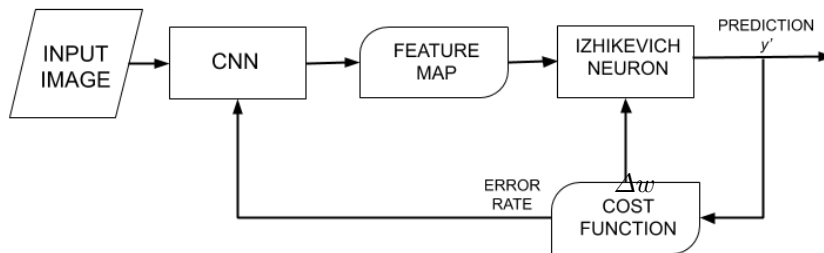


Fig. 1. Network fusion scheme.

3.1 Model Setup

As mentioned earlier, Izhikevich neuron depends on a series of constant parameters. These parameters must be configured in an appropriate way to achieve good classification results, experiments were performed with the configurations proposed by [3] and [15], however the best results were obtained with the configuration proposed in [5]. The parameters used are shown in Table 1.

Table 1. Parameters of the Izhikevich neuron.

Parameter	Value
a	0.03
b	-2
C	100
d	100
v_r	-60
v_t	-40
c	-50
k	1.7
v_{peak}	35
α	1.0^{-5}
w	0.0

Although the original model proposed by Izhikevich establishes the value of the bias θ as a constant, the observation of the antecedent paradigms is that they update this value of bias in a similar way to the weights w , for this reason the experimentation was carried out using a proposed update of θ expressed in

(10), in such a way that it adapts to the data and it is not necessary to do the tuning manually for each case:

$$\theta = \theta + \alpha \cdot (y - y') \cdot (w \times x). \tag{10}$$

The CNN shares some parameters of the Izhikevich neuron as the learning rate α . This proposed model has layers of pooling, in total it has 3 of these, one of max pooling (Equation 4, Section 2.1) with kernel of dimension $N = 3$. The other two consist of layers of average pooling (Equation 5, Section 2.1), of dimensions $N = 4$ and $N = 2$. The justification for this configuration is the reduction of obtained characteristics, since this way a vector of 384 input values for the Izhikevich neuron is achieved. When leaving the configuration without pooling layers, a vector of 588,800 input values is obtained, which can greatly affect the behavior of the pulses. the whole of this proposed model can be seen in Figure 2.

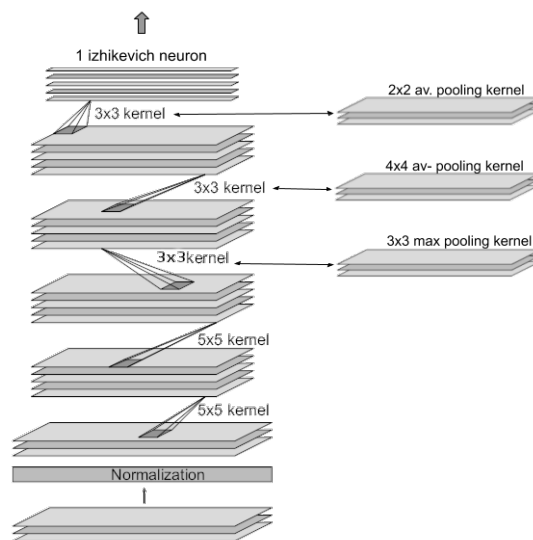


Fig. 2. Proposed model.

As shown in Figure 2, the proposed model is composed of two blocks given by the Convolutional Neuronal Network and Izhikevich neuron. In the CNN block it appears as input an image of size 66×200 with 3 channels (RGB). Layer 1 (C1) convolves the input image with 24 filters of size 5×5 obtaining 24 size feature maps 62×196 without reduction:

$$C1_{i,j} = \sigma \left(\sum_{d=0}^2 \sum_{m=0}^4 \sum_{n=0}^4 w_{m,n}^d \cdot x_{i+m,j+n}^d + b \right). \tag{11}$$

To make the output of the nonlinear linear operation, it is advisable that the convolution output is passed through a Linear Units Rectifier (ReLU) $\sigma(x) = \max(0, x)$. Layer 2 (C2) convolves previous maps with 36 filters of size 5×5 , obtaining 36 feature maps of 58×192 . Layer 3 (C3), the convolution is carried out using 48 filters of 5×5 , obtaining 48 maps of 54×188 . Data reduction is done from layer 4 (P4) of *Max Pooling* (4) with filter size 3×3 getting the same 48 feature maps but with size 18×62 :

$$\mathcal{P}4_{i,j} = \max \begin{pmatrix} \mathcal{C}3_{i,j} & \mathcal{C}3_{i+1,j} \\ \mathcal{C}3_{i,j+1} & \mathcal{C}3_{i+1,j+1} \end{pmatrix}. \quad (12)$$

Layer 5 (C5) convolves through 64 filters of 3×3 , obtaining 64 maps of 16×60 . In layer 6 (P6), the reduction is made through the filter *Average Pooling* (5) of size 4×4 , obtaining 64 maps of 4×15 :

$$\mathcal{P}6_{i,j} = \frac{\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \mathcal{C}5_{i+m,j+n}}{N^2}. \quad (13)$$

Layer 7 (C7) convolves with 64 filters of 3×3 , obtaining 64 maps of size 2×3 . Layer 8 (P8) reduces the size of the maps with a filter *Average Pooling* of 2×2 , obtaining 64 maps of 1×6 . Finally, layer 9 (F) crushes the maps obtained in P8 of size $d \times m \times n$ in a one-dimensional vector, $\mathcal{F} : \mathcal{P}8^{d \times m \times n} \mapsto \mathcal{P}8^{1 \times (d \times m \times n)}$. Given the output of \mathcal{F} , the CNN output vector consists of 384 values.

In the Izhikevich block, x represents the vector \mathcal{F} . The x entry is processed by the operations performed in the Algorithm 1, The output of the block consists of the pulse train $y' \in [0, 100]$ given by the activation function (8), similar to a multilayer perceptron in the discrete domain. The pulse trains are interpreted as the discrete prediction obtained from the proposed model, to evaluate the prediction a transformation processes are performed, which is detailed in Section 4.1. The synaptic weights w is a vector of the same length of the inputs x , through the inner product and the control of a hyperparameter θ calculates the synaptic energy of the neuron, it is accumulated during 1000 iterations generating a pulse when it exceeds the threshold v_{peak} . The total of pulses obtained in the neuron is recognized as the prediction, this serves to calculate the loss function which is used for the adjustment of the weights w (9) and retropropagated to the convolutional block.

The convolutional block is adjusted by the optimization algorithm Adam [11], once the Izhikevich block obtains the error function, it is retropropagated to the previous block to do the adjustment of the synaptic weights in the training iteration i (14), where α is the learning rate, (m_i) the exponential moving average of the gradient (loss function), (v_i) is the squared gradient and ϵ a hyperparameter of the algorithm:

$$w_i \leftarrow w_{i-1} - \frac{\alpha_i \cdot m_i}{(\sqrt{v_i} + \epsilon)}. \quad (14)$$

4 Experimental Setup

4.1 Data

As an experimental purpose, it was proposed to work with images captured in the Udacity simulator, which is used for the development of autonomous vehicles. The information provided by this simulator is a pattern of three images obtained from simulated cameras mounted on the vehicle, directed towards the front of the road, the three cameras remain perpendicular to the vehicle and are distributed to the center and one in each respective position of the edges side of the car, these images are shown in Figure 6. Each image has a resolution of 320×160 pixels and is trimmed to remove 60 pixels containing the sky and 25 pixels containing the front of the vehicle. Coupled with this pattern of three images, there are as a class an angle of the steering wheel, which represents the expected action depending on the shape of the road.

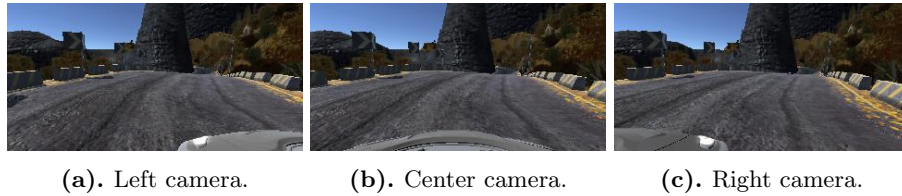


Fig. 3. Images obtained from manual driving in the Udacity simulator.

It is worth mentioning that this angle of rotation of the steering wheel is represented in radians and comprises in a closed interval of $y = [-0.41, 0.41]$ so there is a total of 83 classes if only two floats are considered, as shown in Figure 7. A limitation is that this model has only a single classification node, so a problem of 83 classes is an unfair challenge for a single neuron, and on the other hand increasing the number of Izhikevich neurons would represent a computational cost equal to or greater than the Multilayer Perceptron. Therefore, the Equation (16) is proposed to perform the transformation of this angles to N classes. For example in Figure 4b, the angles were divided into 3 intervals conformed by the expressions mentioned in (12). In the same way it can observe the intervals of 10 classes in the Figure 9. This transformation allows experiments to evaluate the performance of the proposed model as the complexity of the problem rises:

$$f(y) = \left\{ i \text{ if } \left(y_{min} + \frac{|y_{min}-y_{max}|}{classes}i \right) \leq y < \left(y_{min} + \frac{|y_{min}-y_{max}|}{classes}(i+1) \right), \quad (15)$$

$$f(y) = \begin{cases} 0 & \text{if } y < -0.13, \\ 1 & \text{if } -0.13 \leq y < 0.13, \\ 2 & \text{if } 0.13 \leq y. \end{cases} \quad (16)$$

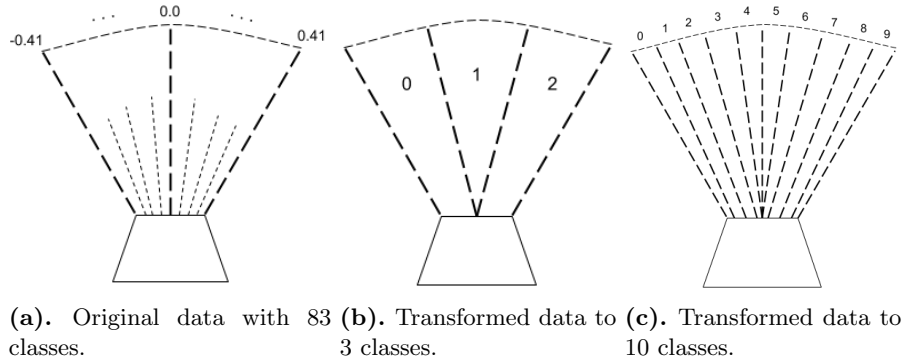


Fig. 4. Steering wheel angle as class.

However, classification with the Izhikevich neuron returns only positive integers, which are defined as pulses. Therefore, the normalization of the data is proposed using (17) to obtain values in the closed interval $y' = [0, 100]$, in order to compare the output pulses with the real classes of the database:

$$y' = \frac{y - y_{min}}{y_{max} - y_{min}} \cdot 100. \quad (17)$$

Now, if the angles (classes) of the database were scaled with (17) to be represented as pulses, both these angles and the output pulses of the neuron need to be rescaled to the interval $y = [-0.41, 0.41]$ to be later interpreted. To perform this scaling process, is used the inverse algebraic of (17), whose expression is shown in (18):

$$y = \frac{y'}{100} \cdot (y_{max} - y_{min}) + y_{min}. \quad (18)$$

The driving database was created with a total of 11,343 instances, which contains a total of 34,029 images. Although the three views are offered, in this experiment it is not the case to predict the exact angle of rotation, for which reason only the image captured by the central camera will be taken into account. For the training parameters, different combinations were made to evaluate the amount of data that can be classified correctly. These experiments involve evaluating both models in training through 50, 100 and 200 epochs to evaluate the learning behavior. The performance test is also carried out in classification, evaluating the classification for 3, 5 and 10 classes, each experiment with their respective validation set equivalent to 20% of the total training lot, using only data that was not used for the training.

4.2 Evaluation

For this experimentation, two types of evaluation are required; The first type is to measure the rate at the proposed model learns from training data and

backpropagation. To perform this measurement it is necessary to graph the values obtained of the cost function through each training epoch. This cost function is given by the Mean Squared Error expressed in (19), the metric obtains the mean of prediction error taking the difference between the expected output and the obtained output. As a support to this metric, it is also proposed to evaluate with the metric of the Root Mean Square Error in (20), observing that one of the classes is negative and the MSE may present variations:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y'_i - Y_i)^2, \quad (19)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y'_i - Y_i)^2}. \quad (20)$$

The second type of evaluation measures the quality of the prediction. Therefore, it is proposed to use the metrics derived from the confusion matrix. The main diagonal represents the true positives (TP), that is, the prediction hits. However, to calculate false positives (FP) it is necessary to add the values of the x column except the value belonging to the TP. For the calculation of false negatives (FN) it is the same procedure but now with the sum of the row x . The first metric derived from the confusion matrix is the recall expressed in (21), this metric measures the percentage of patterns that were correctly classified among all the patterns. The second metric is precision, this metric expressed in (22) is similar to the previous one and measures the percentage of success for a single class. Finally, the F measure metric is shown in (23). This method combines precision and recall to obtain a balanced measure:

$$recall = \frac{TP}{TP + FN}, \quad (21)$$

$$precision = \frac{TP}{TP + FP}, \quad (22)$$

$$Fmeasure = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \quad (23)$$

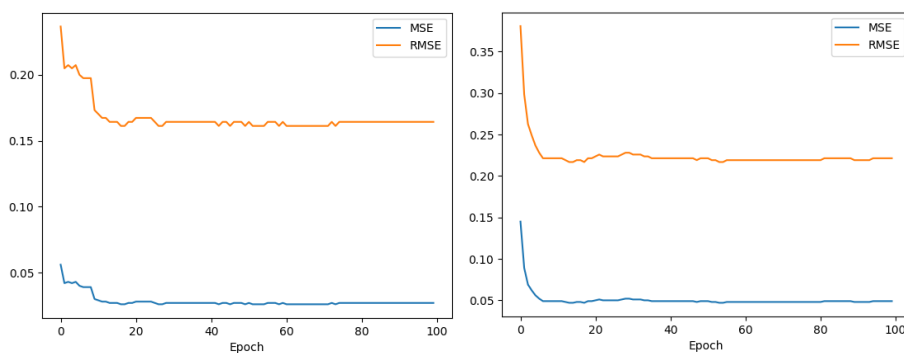
5 Results

The implementation of this fusion was developed in the Python 3.4 language using the Tensorflow 1.1.0 and Keras 2.2 libraries supported by CUDA 8.0 and cuDNN 5.1 drivers for the CNN. The implementation of the Izhikevich neuron was done in Python t0o without the support of the mentioned libraries. The environment used for this experimentation consists of an Intel Core 17-7700 with 4 cores 8 threads 3.6 GHz, SSD storage 120 GB and a GPU Titan X, 3072 cores CUDA 1075 MHz 12 GB RAM. All under the Xubuntu 14.04 Trusty Tar operating system.

In order to compare the experimentation of the proposed model, the deep neural model Pilotnet [1,2] was evaluated. This model was designed to train with the 83 classes mentioned in Section 4.1, but it was evaluated in the same way as the proposed model.

5.1 Training Results

As mentioned in the Section 4.2, the experimentation was carried out under the criteria of training and classification quality. For this first experiment, the MSE and RMSE cost metrics were plotted to measure the rhythm of learning of the proposed model. In short, the graph should show a slope that tends to 0, it indicates that a better classification is being done in the validation stages after a training period. In Figure 13 shows the comparison of the gradients obtained in the training of 100 epochs.



(a). Training with 100 epochs and 5 classes. (b). Training with 100 epochs and 3 classes.

Fig. 5. Mean Square Error and Mean Quadratic Error obtained during the training of the proposed model.

The gradients obtained in the previous experiment are acceptable since they trend to 0. However, by zooming in on the graph, an erratic behavior in the rhythm of learning is appreciated. These variations depend to a large extent on the dispersion and balance of the classes. Although the gradient is maintained or grows, it remains in acceptable values to make a good classification. The erratic behavior mentioned above can be seen in Figure 14, where a comparison is also made with the gradient of the Pilotnet model.

Observing the graph of the gradient, a great difference can be seen between the learning of both neuronal models. On the other hand, it is appreciated that the proposed model reaches a minimum learning point in which despite adding training epochs this does not improve the results. This observation does not represent a disadvantage, this is an advantage to demonstrate that this neuronal

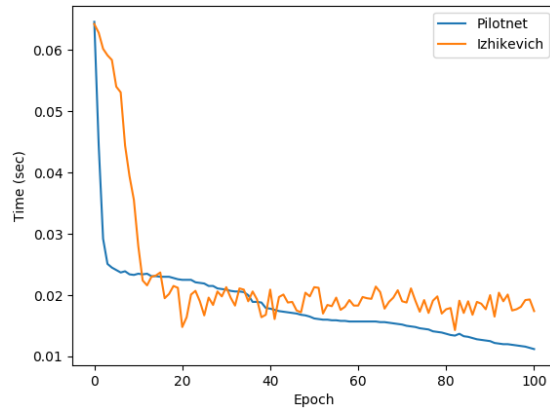


Fig. 6. Cost function in training with 100 epochs and 10 classes.

paradigm requires fewer training periods, however it is limited by capacity and data. Now, the training time is clearly shorter in the proposed model. The time that take to train in the experiments are shown in Figure 15.

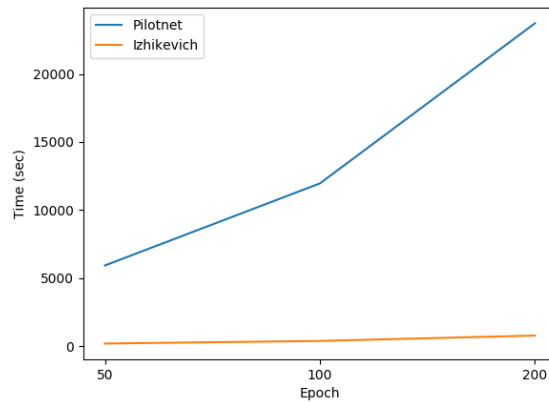


Fig. 7. Time of training for both models.

The time graph shows a big difference in the time it takes to train both models. This is due to the reduction of complexity with respect to the comparison model. To better visualize the time obtained in the experiments, Table 2 is shown, where the time is in seconds for each experiment.

Table 2. Time that took the experiments.

Model	Training Time (sec)
Proposal 50 epochs	189.19
Pilotnet 50 epochs	5,925.55
Proposal 100 epochs	378.53
Pilotnet 100 epochs	11,954.79
Proposal 200 epochs	774.40
Pilotnet 200 epochs	23,719.89

5.2 Classification Results and Comparison

Given the reference, the experimentation was carried out equitably with 100 epochs of training for both models. This evaluation is done for 3, 5 and 10 classes with the intention of evaluating the maximum capacity of the Izhikevich neuron for the classification. The results obtained are shown in Table 3.

Table 3. Results obtained for both models in classification with 100 epochs of training.

Model	Metrics		
	Precision	Recall	F-Measure
Proposal 3 classes	0.991	0.983	0.987
Pilotnet 3 classes	0.999	0.994	0.996
Proposal 5 classes	0.855	0.845	0.849
Pilotnet 5 classes	0.991	0.966	0.966
Proposal 10 classes	0.740	0.741	0.740
Pilotnet 10 classes	0.911	0.897	0.903

Although the model that classifies through the multilayer perceptron presents slightly results, this considerably increases the training time. The proposed model requires much less time of the referent model, even though the performance results in classification are very close in case of 3 classes. Although in the case of increasing the number of classes reduce its classification capacity, depending on the problem to be treated, it could be considered as a faster option. Recalling that this generation of neural networks are in research for new applications.

Another aspect to consider is that the Izhikevich neuron depends on a series of constant values. Although for most of these constants there is a recommended value, in the specific case of the threshold (θ) it is observed that for most of the neuronal paradigms this parameter must be adjustable as the weights (w). As a proposal to contribute to the development of this neuronal paradigm, an update of the threshold is proposed given the function expressed in (10). It is worth mentioning that the experimentation reported here was implemented considering the aforementioned update and showing good results.

6 Conclusions and Future Improvements

As demonstrated in this research, one way to reduce the training time of a Neural Network is to use an alternative to the Multilayer Perceptron Network commonly used for classification. This simple but effective solution is able to reduce training time considerably due to the remarkable decrease in neurons to be processed. As presented in this article, the Izhikevich neuron is able to correctly classify the patterns obtaining metrics very close to those of the reference model in a problem of medium complexity with 3 classes. Although in the literature is reported that this type of neuron turns out to be useful for binary nonlinear classification, thanks to the adjustment of the Convolutional Neuronal Network it can improve the results allowing classifying with more classes.

Although the experimentation shown here consists of not extensive training sets with the full number of classes, the improvement of this proposed model will be able to result in problems of classification in Big Data, which is planned as a future work. Similarly, the dynamic adjustment of the parameters of the Izhikevich neuron, such as the one shown in the Section 3.1 Equation (10), is an aspect to be improved since it depends to a large extent on experiments prior, and as is well known, a neural network is a non-parametric algorithm.

Acknowledgment. The authors express their gratitude to CONACYT, as well as Tecnológico Nacional de México/CENIDET for the support given to the department of computational sciences, in which it was possible to carry out this investigation that is part of the project 5628.19-P so-called "Sistema embebido para asistencia de conducción basado en Lógica Difusa Tipo-2".

References

1. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
2. Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U.: Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911 (2017)
3. Demirkol, A.S., Ozoguz, S.: A low power vlsi implementation of the izhikevich neuron model. In: 2011 IEEE 9th International New Circuits and systems conference. pp. 169–172. IEEE (2011)
4. van Grinsven, M.J., van Ginneken, B., Hoyng, C.B., Theelen, T., Sánchez, C.I.: Fast convolutional neural network training using selective data sampling: Application to hemorrhage detection in color fundus images. IEEE transactions on medical imaging 35(5), 1273–1284 (2016)
5. Hernández-Becerra, M.M.L.: Clasificación de patrones mediante el uso de una red neuronal pulsante. Congreso Mexicano de Inteligencia Artificial 5 (2016)
6. Iakymchuk, T., Rosado-Muñoz, A., Guerrero-Martínez, J.F., Bataller-Mompeán, M., Francés-Víllora, J.V.: Simplified spiking neural network architecture and stdp learning algorithm applied to image classification. EURASIP Journal on Image and Video Processing 2015(1), 4 (2015)

7. Iandola, F.N., Moskewicz, M.W., Ashraf, K., Keutzer, K.: Firecaffe: near-linear acceleration of deep neural network training on compute clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2592–2600 (2016)
8. Izhikevich, E.M.: Simple model of spiking neurons. *IEEE Transactions on neural networks* 14(6), 1569–1572 (2003)
9. Izhikevich, E.M.: Which model to use for cortical spiking neurons? *IEEE transactions on neural networks* 15(5), 1063–1070 (2004)
10. Jerry, M., Chen, P.Y., Zhang, J., Sharma, P., Ni, K., Yu, S., Datta, S.: Ferroelectric fet analog synapse for acceleration of deep neural network training. In: 2017 IEEE International Electron Devices Meeting (IEDM). pp. 6–2. IEEE (2017)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
12. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *nature* 521(7553), 436 (2015)
13. Matsubara, T., Torikai, H., Hishiki, T.: A generalized rotate-and-fire digital spiking neuron model and its on-fpga learning. *IEEE Transactions on Circuits and Systems II: Express Briefs* 58(10), 677–681 (2011)
14. Pattanayak, S.: *Pro Deep Learning with TensorFlow*. Springer (2016)
15. Rice, K.L., Bhuiyan, M.A., Taha, T.M., Vutsinas, C.N., Smith, M.C.: Fpga implementation of izhikevich spiking neural networks for character recognition. In: 2009 International Conference on Reconfigurable Computing and FPGAs. pp. 451–456. IEEE (2009)
16. Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. In: *Artificial Neural Networks–ICANN 2010*, pp. 92–101. Springer (2010)
17. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. *arXiv preprint arXiv:1708.07120* (2017)
18. Zhang, J.R., Zhang, J., Lok, T.M., Lyu, M.R.: A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied mathematics and computation* 185(2), 1026–1037 (2007)

Detección y rastreo de peatones empleando vectores de características de histogramas de gradientes orientados y patrones binarios locales en una máquina de soporte vectorial

Raymundo Rodríguez Romero¹, Luis Pastor Sánchez Fernández¹,
Edgardo Manuel Felipe Riverón², José Juan Carbajal Hernández¹

¹ Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México
ing.m.rodriguez@gmail.com, {edgardo,lsanchez}@cic.ipn.mx, carbajalito@hotmail.com

Resumen. La detección e identificación de peatones, así como su rastreo, es un problema abierto en el área de la visión artificial que se ha atacado de formas muy diversas. Este trabajo presenta una propuesta para la detección de peatones utilizando técnicas de análisis de imágenes para la extracción de características de un peatón basadas en histograma de gradientes orientados y patrones binarios locales los cuales, son extraídos de una región de interés previamente detectada, combinados e ingresados a una máquina de soporte vectorial que funge como clasificador entre muestras de peatones y no peatones. Un rastreador basado en el algoritmo *MeanShift* es implementado por su eficiencia en el procesamiento del rastreo y la identificación de las nuevas ubicaciones de cada uno de los objetivos detectados.

Palabras clave: detección de objetos, gradientes orientados, patrones binarios, MeanShift, soporte vectorial.

Detection and Tracking of Pedestrians using Feature Vectors of Oriented Gradient Histograms and Local Binary Patterns using a Support Vector Machine

Abstract. The detection and identification of pedestrians, as well as their tracking, is an open problem in the area of artificial vision that has been worked in many different ways. This paper presents a proposal for the detection of pedestrians using image analysis techniques for the extraction of characteristics of a pedestrian based on histogram of oriented gradients and local binary patterns which are extracted from a region of interest previously detected, combined and entered to a vector support machine that functions as a classifier between samples of pedestrians and non-pedestrians. A tracker based on the MeanShift algorithm is implemented for its efficiency in tracking processing and identifying the new locations of each of the detected targets.

Keywords: object detection, oriented gradients, binary patterns, MeanShift, vector support.

1. Introducción

La detección y el rastreo de objetos han representado un reto en el área de la visión artificial desde hace varias décadas, siendo actualmente de los más estudiados gracias al desarrollo de hardware más potente que permite un mejor desempeño en las tareas necesarias para llevar a cabo la detección de los objetos de interés y su rastreo.

En este trabajo se aborda una propuesta de solución a la detección de objetos que excluye aquellos que no correspondan con la descripción de un peatón y posteriormente llevar a cabo el rastreo del objetivo utilizando una caja de contenido.

La detección de objetos en una toma aleatoria ha sido estudiada empleando diferentes algoritmos, generalmente sustrayendo el fondo con diversas técnicas de análisis de imágenes y modelos de color, por ejemplo, Leykin [9] presentó una propuesta en el 2010 empleando un modelo basado en la temperatura del color para la sustracción del fondo y la detección de los objetos visibles. Por otro lado, Zheng et al. [1] trabajó en el 2017 una propuesta de actualización continua empleando un modelo de color RBG y aplicando promedios sobre cada pixel, de forma tal que, al evaluar una imagen con un objeto móvil, dicho modelo genere una diferencia probabilística sobre los pixeles en el fotograma evaluado. En el 2016 Lacabex et al. [5] realizó la detección de objetos en su sistema empleando una resta directamente entre el fondo y los fotogramas actuales en un modelo RGB, aplicando posteriormente dos umbrales en cada canal de color: uno bajo y uno alto; tras lo que aplicó una operación de correlación entre ambos.

Existen otros trabajos que no realizan una búsqueda de objetos en la imagen, sino que evalúan directamente la existencia de peatones, como en el caso de Yi et al. [2] donde se empleó una búsqueda por contorno y formas a través de un algoritmo de máquina de estados algorítmica (ASM) [11] y un modelo probabilístico basado en una distribución Gaussiana.

Lim y Kim presentaron en el 2013 una detección de peatones basada en la extracción de características empleando el histograma de gradientes orientados [6] (*Histogram of Oriented Gradient*, HOG por sus siglas en inglés) para posteriormente realizar la clasificación utilizando el algoritmo AdaBoost propuesto por Viola y Jones [10]. Por su parte, Li et al. [7] realizaron un trabajo con algunas similitudes al de Lim et al., sin embargo, la principal diferencia fue el uso de una máquina de soporte vectorial (*Support Vector Machine*, SVM por sus siglas en inglés) las cuales se basan en la construcción de hiperplanos que separan las muestras en clases linealmente separables utilizando los llamados vectores de soporte.

Existen diversas propuestas para realizar el rastreo de un objetivo; entre aquellas que se basan en peatones tenemos ejemplos como: el trabajo de Chen et al. [8] en el cual el rastreo se realiza en cada fotograma con la detección continua de objetivos y la diferencia entre ambos para determinar las nuevas ubicaciones a través de una detección de orientación de movimiento. Por otra parte, el trabajo de Li et al. [7] también incluye un rastreador basado en una combinación entre los algoritmos de rastreo Camshift [13] y el filtro Kalman.

En nuestra propuesta, se aplica una separación de los canales de color empleando los canales de intensidad, luminiscencia y los tres canales RGB extrayendo la información útil de cada canal, realizando un umbralado y sumándola posteriormente para obtener los objetos existentes en la toma.

Además, se empleó una clasificación entre peatones y no peatones utilizando un clasificador SMV, con la diferencia de que es alimentado por vectores de características HOG y patrones locales binarios (*Local Binary Patterns*, LBP por sus siglas en inglés) [4] los cuales describen las características de textura en una imagen. Finalmente se utilizó un algoritmo *MeanShift* [12] para llevar a cabo el rastreo de los peatones encontrados, basado en la búsqueda por ventana de histograma de color que describan a los objetivos y su centroide.

2. Metodología

El modelo de solución se basa en diversas etapas secuenciales que corresponden a los procesos necesarios para segmentar las características, de forma que el sistema pueda detectar y rastrear a los objetivos en los fotogramas procesados mostrado en la Fig. 1.

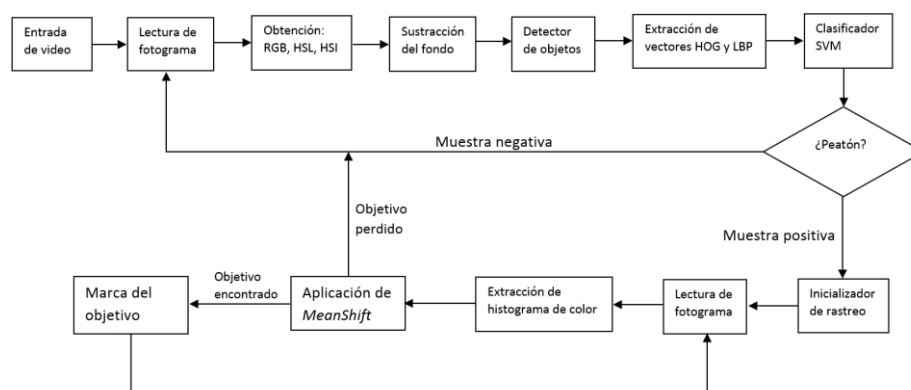


Fig. 1. Diagrama de flujo del algoritmo de rastreo.

2.1. Detección de peatones

Empleando la lectura de los fotogramas de video se obtienen los canales HSL, HSI y RGB (Fig. 2) mediante los cuales se realizan diversas operaciones de análisis de imágenes y de morfología matemática para la detección de peatones.

El modelo propone tener siempre la información sobre el fondo de la imagen, de forma que se pueda optimizar el proceso de detección de objetos.

Para identificar diferentes características en la toma de acuerdo a los cambios de luz y color de los objetos presentes en la toma extraemos los canales de luminiscencia, intensidad, RGB y su conversión a escala de grises.



Fig. 2. a) Fondo RGB. b) Fondo a escala de grises. c) Fondo en canal de luminiscencia. d) Fondo en canal de intensidad.

Se emplea una operación para obtener la media del fondo en cada uno de los canales propuestos cuando no se han detectado objetivos para el rastreo (1), lo que permite la reducción del ruido generado por cambios de iluminación:

$$\mu_t = \frac{\sum_{j=0}^n F_j}{n}, \quad (1)$$

donde μ_t representa la media del fondo a través del tiempo, $\sum_{j=0}^{j-1} F_j$ representa la sumatoria del conjunto de fotogramas analizados donde no se ha detectado ningún objeto y n es el número total de fotogramas analizados que cumplen esta condición.

Para realizar la detección de los objetos en el fotograma actual se sustrae la media del fondo a la imagen actual, aplicamos valor absoluto al resultado (2) y finalmente se umbrala el contenido de la imagen obteniéndose una imagen binaria (3):

$$T_j = |F_j - \mu_t|, \quad (2)$$

$$T_j = \{255 \text{ Si } p(x, y) \geq u \text{ 0 Si } p(x, y) < u, \quad (3)$$

donde F_j es el valor del fotograma actual, μ_t define el promedio del fondo, T_j es el valor del fotograma resultado de la resta, mientras que $p(x, y)$ representa cada uno de los pixeles presentes en T_j .

Aplicando la Ec. (3), obtenemos objetos más cercanos a los presentes dentro de la escena en cada uno de los canales de color procesado, los cuales serán almacenados en un único fotograma realizando la unión lógica de las 4 imágenes binarias como se observa en la Fig. 3, donde A_j , B_j , C_j , y D_j son los planos binarios y U_j es la unión de todos ellos (4).

$$U_i = A_j \cup B_j \cup C_j \cup D_j. \quad (4)$$

Para eliminar las partículas que no son parte de los objetivos detectados sino pequeños segmentos de ruido presentes en la toma, utilizamos un operador morfológico de apertura (7),

$$A_D = A \oplus B, \quad (5)$$

$$A_E = A \ominus B, \quad (6)$$

$$A \circ B = (A \oplus B) \ominus B, \quad (7)$$

donde A representa al conjunto de píxeles de la imagen procesada, B es el elemento de estructura seleccionado (un cuadrado de 3x3 píxeles), La Ec. (5) muestra la operación de dilatación del elemento de estructura sobre la imagen binaria y la Ec. (6) representa la operación de erosión del elemento de estructura sobre la imagen dilatada.

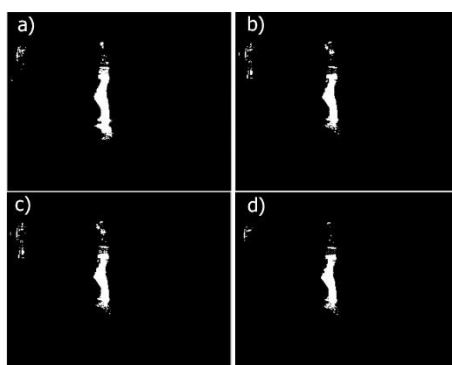


Fig. 3. a) Objetos obtenidos desde RGB. b) Objetos obtenidos desde la escala de grises. c) Objetos obtenidos desde el canal de luminiscencia. d) Objetos obtenidos desde el canal de intensidad.

Finalmente, la operación de dilatación extra es aplicada con el fin de ampliar los objetos detectados y tener una mejor percepción de ellos durante la selección de la región de interés, el resultado de estas operaciones se puede observar en la Fig. 4.

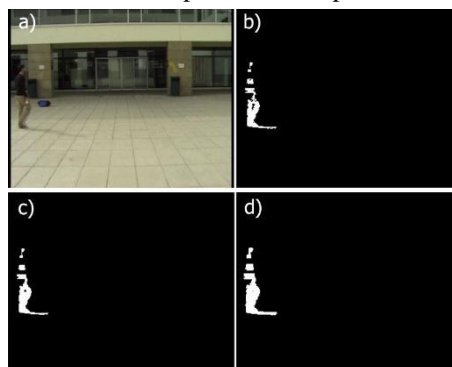


Fig. 4. a) Fotograma actual b) Unión de imágenes binarias c) Apertura morfológica. d) Dilatación morfológica con elemento de estructura cuadrado de 3x3 píxeles.

Con los datos obtenidos tras las operaciones morfológicas se tiene una descripción de los objetos detectados, lo cual permite obtener regiones de interés (*Region of interest*, ROI, por sus siglas en inglés) sobre las cuales se determina si han sido o no identificados los peatones entrantes a la toma. Las ROI se utilizan para analizar los

objetos detectados y extraer sus características mediante vectores dados por el histograma de gradientes orientados y los patrones binarios locales.

2.1.1. Extracción de características HOG

El histograma de gradientes orientados se basa en el supuesto de que toda imagen tendrá los mismos cambios en su iluminación sin importar si ésta es clara u oscura, además, tiene un funcionamiento eficiente cuando se evalúan imágenes con cambio de escala; gracias a que otorga una descripción de los gradientes dentro de la imagen y la dirección de estos.

Dada una imagen en escala de grises dividida en celdas (para este caso contempladas como 8x8) se determina el gradiente computacional descrito por las Ecs. (8) y (9):

$$G_x(x, y) = [-1, 0, 1] \cdot [I(x-1, y), I(x, y), I(x+1, y)]^T, \quad (8)$$

$$G_y(x, y) = [-1, 0, 1] \cdot [I(x, y-1), I(x, y), I(x, y+1)]^T. \quad (9)$$

La magnitud de cada pixel está definida por la ec. (10):

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \quad (10)$$

mientras su dirección está dada de la siguiente forma (11):

$$\varphi(x, y) = \arctan \left[\frac{G_y(x, y)}{G_x(x, y)} \right]. \quad (11)$$

Convencionalmente se utilizan 9 contenedores que dispondrán de todos los ángulos entre 0 y 180 grados. Cuando el ángulo de un pixel entra dentro del rango del contenedor, la magnitud de este es sumada al mismo y contemplada como 0 para todos los demás, tal y como lo describe la Ec. (12):

$$hog_i = \{G(x, y)\varphi(x, y) \in contenedor(i) \mid 0 \notin contenedor(i)\}. \quad (12)$$

Agregamos cada uno de los valores obtenidos a los vectores de características de cada una de las celdas dentro de toda la imagen. Para poder tener una descripción correcta de todos los cambios de iluminación y contraste, es necesario normalizar localmente, por lo que debemos agrupar las celdas en grupos de mayor tamaño. Dado lo anterior, el descriptor por bloques está dado por la concatenación de los histogramas normalizados generados en cada celda. Estos bloques generalmente suelen sobreponerse lo que significa que cada una de las celdas podrá ser considerada en más de una ocasión para el descriptor final (Ec. 13):

$$hog_B = \frac{hog_B}{\sqrt{\|hog_B\|^2 + \epsilon}}, \quad (13)$$

donde hog_B es el descriptor de cada bloque y ϵ representa un pequeño valor constante (teóricamente no importa cuál) agregado para evitar problemas numéricos en la raíz cuadrada.

Entonces, el descriptor final estará dado por la concatenación de cada uno de los bloques dentro de los 9 contenedores.

2.1.2. Extracción de características LBP

Por su parte, LBP describe las características de textura encontradas en un objeto analizando los píxeles de forma local y otorga valores de peso en aquellos que tengan una relación de valor muy cercana y a su vez aleja otros que no tengan ninguna relación o un parentesco.

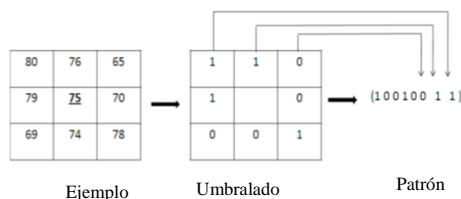


Fig. 5. Cálculo de LBP.

Acorde al ejemplo anterior (Fig. 5) podemos determinar el valor del pixel central con una máscara de 3x3 realizando un barrido con sus 8 vecinos otorgando un peso binario a cada uno de ellos, con lo que para el caso anterior 10010011 estaría definido como $LBP = (10010011)_2 = 147$.

De esta forma podemos definir la textura de una imagen como la distribución conjunta de sus niveles de gris en una área local definida por la equidistancia entre el pixel central (R) y los vecinos contemplados (P) (Fig. 6) que está dada por la Ec. (14).

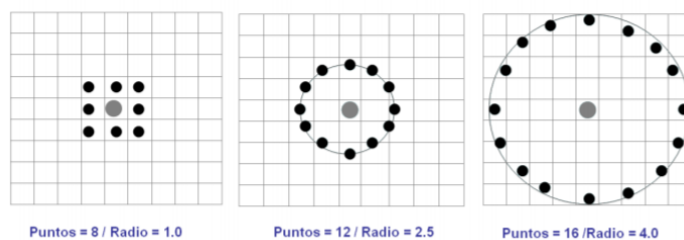


Fig. 6. Simetría circular para conjuntos de vecinos [3].

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad (14)$$

donde s es una transformación definida por el signo derivado de la diferencia entre g_c y g_p los cuales representan al pixel central evaluado y a cada uno de los píxeles vecinos contemplados para la transformación LBP respectivamente, mientras que 2^p es el peso asignado a cada uno de los vecinos.

Una vez obtenida la nueva imagen definida por la transformación LBP se aplican mallas en las direcciones X y Y , y se divide la imagen en secciones dentro de las cuales obtendremos los histogramas LBP que serán utilizados como entradas para las SVM.

2.1.3. Clasificación utilizando SVM

Las máquinas de soporte vectorial (SVM) son clasificadores que permiten trabajar con muestras linealmente separables, como en nuestro caso se tienen dos tipos de

objetos (peatón o no peatón). Este tipo de clasificador nos permite separar el problema muy fácilmente con las muestras adecuadas.

El clasificador está dado por la relación de las muestras etiquetadas (Ec. 15):

$$\psi_{1:n} = \{(h_1, y_1), (h_2, y_2), \dots, (h_n, y_n)\}, \quad (15)$$

donde $(h_i, y_i) \in \psi_{1:n}$ y $y_i \in \{-1, 1\}$; el conjunto $\psi_{1:n}$ es linealmente separable si existe un vector que satisfaga la Ec. (16):

$$y_i(w \otimes h_i + b) \geq 1. \quad (16)$$

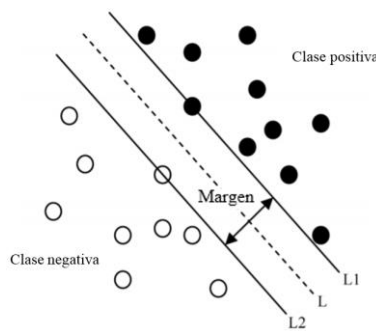


Fig. 7. Clasificador máquina de soporte vectorial de dos clases [7].

Dado lo anterior y como se observa en la Fig. 7 se puede construir una matriz de confusión (Tabla 1) que describa el problema de clasificación en las muestras evaluadas por la red neuronal una vez realizado el entrenamiento con muestras positivas.

Tabla 1. Matriz de confusión de la SVM para la detección de peatones.

Resultado de la clasificación	Muestra – No peatón	Muestra – Peatón
No peatón	Verdadero Negativo	Falso Positivo
Peatón	Falso Negativo	Verdadero Positivo

Para el entrenamiento de la SVM se utilizan los vectores de características descritos anteriormente, realizando una concatenación entre ambos descriptores para obtener un solo vector de características, el cual contiene toda la información sobre las muestras ingresadas a la red neuronal. La Fig. 8 muestra las ROI identificadas en el objeto, las cuales nos permiten identificar las zonas para el análisis de la SVM.



Fig. 8. a) Objetos encontrados para el análisis de las ROI. b) Peatón detectado con los descriptores HOG y LBP.

2.2. Rastreo de peatones

Para realizar el rastreo, una vez encontrados los objetivos que cumplen con las características de ser un peatón, se utiliza un algoritmo *MeanShift*, que se basa en la estimación del kernel de densidad en el histograma de color, el cual está conformado por los niveles obtenidos en el canal de matiz (*Hue* o *H* por su sigla en inglés) de la imagen, para este caso se optó por emplear 64 contenedores para optimizar la velocidad de procesamiento.

Se emplea un fotograma anterior para determinar la posición en el fotograma actual a través de un análisis de la densidad del kernel, el cual se basa en la obtención de los valores de distancia (ponderados) en un área de análisis circular a partir de una posición central determinada aleatoriamente en un área de búsqueda en una posición cercana a la original. De esta forma determina el centroide de la densidad actual, lo que muestra el desplazamiento del objetivo entre un fotograma y otro.

Se obtiene la nueva ubicación del objetivo aplicando iteraciones hasta encontrar el máximo local dentro de la nueva imagen y asignando la nueva posición dada por la Ec. (17),

$$\Delta P = \frac{\sum K(P - P_{ant})w(P)(P - P_{ant})}{\sum K(P - P_{ant})w(P)}, \quad (17)$$

donde, P_{ant} y P representan la posición anterior y actual respectivamente y w describe el cambio en la región de interés del candidato y el objetivo generado por los histogramas de color de la Ec. (18),

$$w(P) = \sqrt{\frac{h_{obj}(I(P))}{h_c(I(P))}}, \quad (18)$$

donde h_{obj} y h_c son los histogramas de color del objetivo y del candidato, respectivamente.

La Fig. 9 muestra cuando un objetivo es detectado dentro de la ventana de búsqueda, la cual actualiza la posición de la caja de contenido y avanza al siguiente fotograma para verificar que no existan nuevos objetivos dentro de la toma.



Fig. 9. Rastreo de un objetivo con cambios de iluminación.

3. Resultados experimentales

Las distintas secciones del algoritmo fueron probadas por separado y de forma completa con la base de datos EPFL *Multi-camera pedestrian video* [], ya que dispone de diferentes entornos para la comprobación de su funcionamiento tanto en iluminación como en texturas y tipos de fondo. Cada escena fue grabada con 3 o 4 cámaras con diferentes ubicaciones de forma paralela, a una altura de 2 metros desde el suelo; los peatones son personal del laboratorio de visión por computadora de la Escuela Politécnica Federal de Lausana, mismo que realizó las grabaciones y compartió la base de datos.

El sistema de rastreo desarrollado es capaz de detectar objetos en cualquier tipo de entorno ya que tiene gran robustez ante cambios de iluminación gracias a que el uso de los canales de luminiscencia e intensidad proporcionan las características de luz ambiental en los fotogramas analizados, mientras que la integración RGB brinda la información necesaria para actualizar el cambio de color para cada región de interés encontrada. La Fig. 10 muestra el detector de objetos aplicado en dos áreas diferentes, una solo iluminada por luz artificial, mientras que en la otra se dispone únicamente de iluminación natural. En ambos casos los objetos presentes son extraídos de forma exitosa.

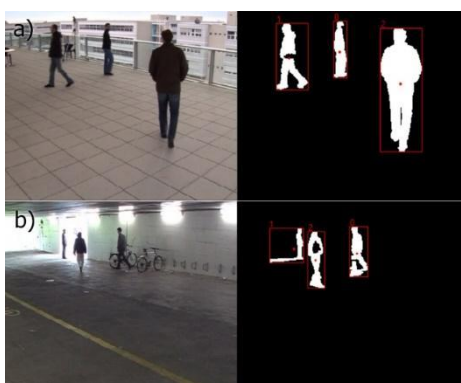


Fig. 10. a) Detección en entorno con luz natural. b) Detección en entorno con luz artificial.

El uso de descriptores de características tales como HOG y LBP es muy común en el área de la visión artificial, ya que los alcances para sus respectivas aplicaciones

pueden ser muy amplios. La combinación de ambos descriptores como entrada para un clasificador tipo SVM (Tabla 2) mejoró en gran medida la detección de falsos positivos en comparación con el uso de un solo descriptor como entrada al clasificador como se observa en la Fig. 11.

Tabla 2. Precisión del clasificador SVM con diferentes características de entrada.

Extractor de características	Precisión	Falsos negativos	Falsos positivos
HOG	89%	5%	6%
LBP	78%	14%	8%
HOG y LBP	92%	6%	2%

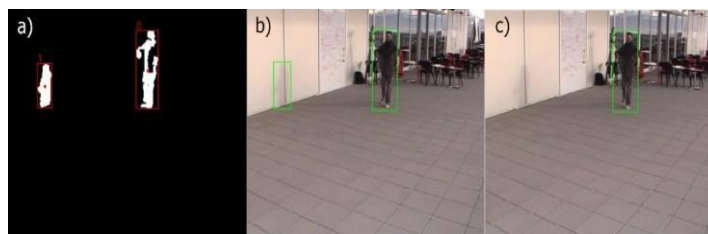


Fig. 11. a) Detección de objetos. b) Falso positivo detectado con HOG. c) Detección de objetivo con HOG y LBP.

La precisión del algoritmo se calculó realizando una detección de peatones en un conjunto de 100 tomas en las cuales ya se tenía identificada la cantidad de peatones presentes en cada toma, se considera correcto solo el caso donde se identifica un peatón correctamente, un falso negativo es el caso donde un peatón no es identificado como tal y un falso positivo es el caso en que un objeto es identificado como peatón (comúnmente dado en sombras).

El algoritmo *MeanShift* tiene muy buen funcionamiento cuando los objetivos se encuentran en entornos contrastantes; sin embargo, tiende a perderlos cuando estos entran en zonas donde la iluminación puede confundirlo con el fondo. En casos extremos incluso se puede perder definitivamente, lo que originaría una nueva búsqueda desde la etapa de detección.

La Fig. 12 muestra la secuencia de rastreo de un objetivo con un fondo iluminado por luz natural, la precisión obtenida en cada una de las secuencias puede verse reflejada en la Tabla 3.



Fig. 12. Rastreo de un objetivo único.

Tabla 3. Porcentaje de rastreo exitoso en escenas con un solo objetivo.

Secuencia	Precisión
1	93%
2	92%
3	95%

En la Fig. 13 se aprecia el rastreo de varios peatones, apreciando el rastreo correcto cuando se encuentran distanciados y la pérdida por oclusión en otros casos, la precisión obtenida en cada secuencia para este caso puede ser consultada en la Tabla 4.

Se probaron 3 tipos de escenas diferentes de 300 fotogramas para determinar la precisión del rastreo, contemplando el seguimiento desde la entrada hasta la salida de cada objetivo sin la existencia de una nueva detección.



Fig. 13. Rastreo multiobjetivo.

Tabla 4. Porcentaje de rastreo exitoso en escenas con varios objetivos

Secuencia	Precisión
1	92%
2	90%
3	93%

Este tipo de rastreo permite la ubicación de los objetivos de forma eficiente; sin embargo, al ser dependiente del histograma de color, las oclusiones o encuentros entre objetivos tienden a afectar en gran medida la nueva ubicación de las posiciones, ocasionando pérdida o confusión entre objetivos.

4. Conclusiones y trabajo futuro

La detección de objetos empleando una metodología simbólica en combinación con técnicas de análisis de imágenes, tuvo como resultado un sistema de detección de objetos eficaz y robusto ante cambios de iluminación, lo que permite utilizar el algoritmo en cualquier tipo de fondo, ya sea con iluminación natural o artificial.

Por otro lado, los descriptores de imágenes basados en HOG y LBP permiten la invariancia al nivel de gris y al escalamiento, lo que los ha llevado a ser utilizados

principalmente en la detección de rostros o tipos de texturas; la propuesta ayuda a mejorar la precisión para determinar la existencia de un peatón sobre cualquier tipo de iluminación y excluir objetos que no deben ser rastreados (como automóviles, animales, sombras, etc.). Debido al nivel de confianza que garantiza el clasificador SVM se puede optar por utilizar un algoritmo de rastreo simple que no consume grandes recursos computacionales como lo es *MeanShift*.

MeanShift tiene grandes ventajas, tales como un procesamiento rápido basado en el histograma de color y por lo tanto aprovecha toda la información que brinda el modelo RGB. También la búsqueda se basa en un entorno local, lo cual facilita la ubicación de la nueva posición. A pesar de esto, el rastreo tiene puntos a mejorar como la pérdida por oclusión o la confusión de objetivos al confundir dos objetivos con un histograma de color muy parecido.

El sistema puede mejorarse con pruebas de otros algoritmos de rastreo y la inclusión de bibliotecas para el reconocimiento, con el fin de no perder aquellos objetivos que hayan sido detectados al menos una vez anteriormente.

Referencias

1. Zheng, L., Ruan, X., Chen, Y., Huang, M.: Shadow removal for pedestrian detection and tracking in indoor environments. *Multimedia Tools and Applications* 76(18), 18321–18337 (2017)
2. Yi, X., Mingtao, P., Jiagen, Z., Meng, M., Yunde, J.: Tracking Pedestrian with Incrementally Learned Representation and Classification Model. *Journal of Information Science and Engineering* 30(4), 1035–1052 (2014)
3. Rami, H., Hamri, M., Masmoudi, L.: Objects Tracking in Images Sequence using Local Binary Pattern (LBP). *International Journal of Computer Applications* 63(20), 19–23 (2013)
4. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distribution. *Pattern Recognition* 29(1), 51–59 (1996)
5. Lacabex, B., Cuesta-Infante, A., Montemayor, A., Pantrigo, J.: Lightweight tracking-by-detection system for multiple pedestrian targets. *Integrated Computer-Aided Engineering* 23(3), 299–311 (2016)
6. Lim, J., Kim, W.: Detecting and tracking of multiple pedestrians using motion, color information and the AdaBoost algorithm. *Multimedia Tools and Applications* 65(1), 161–179 (2013)
7. Li, D., Xu, L., Goodman, E., Xu, Y., Wu, Y.: Integrating a statistical background- foreground extraction algorithm and SVM classifier for pedestrian detection and tracking. *Integrated Computer-Aided Engineering* 20(3), 201–216 (2013)
8. Chen, Y., Chen, N., Zhou, Y., Wu, K., Zhang, W.: Pedestrian Detection and Tracking for Counting Applications in Metro Station. *Discrete Dynamics in Nature and Society*, pp. 1–11 (2014)
9. Leykin, A., Hammoud, R.: Pedestrian tracking by fusion of thermal-visible surveillance videos. *Machine Vision and Applications* 21(4), 587–595 (2010)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. *CVPR*, pp. 511–518 (2001)
11. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: *Proceedings of International Conference of Computer Vision*, pp. 734–741 (2003)
12. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(5), 564–577 (2003)

Raymundo Rodríguez Romero, Luis Pastor Sánchez Fernandez, et al.

13. Bradski, G.: Computer Vision Face Tracking as a Component of a Perceptual User Interface. In: IEEE Workshop Applications of Computer Vision, pp. 214–219 (1998)
14. Baqué, P., Fua, P.: Multi-camera pedestrians video – CVLAB. Lausana, Suiza: Multi-camera pedestrians video. <https://cvlab.epfl.ch/data/data-pom-index-php/> (Recuperado el 10 de agosto del 2019)

Diseño e implementación de un sistema para determinar el impuesto predial a través del procesamiento y análisis de imágenes satelitales

Roman Alejandro Hernández-Moreno, Jessica Gabriela Azpeitia-Rodríguez,
Derlis Hernández-Lara, Leonardo Miguel Moreno-Villalba

Tecnológico Nacional de México,
Tecnológico de Estudios Superiores de Ecatepec,
Ecatepec, Estado de México, México
{derlis392}@hotmail.com, {jessy.lucis715, shisuka.ari}@gmail.com

Resumen. Este trabajo presenta la forma de implementar recursos de análisis y procesamiento de imágenes para obtener un registro de los inmuebles no actualizados y registrados ante el registro público de la propiedad y comercio, con el fin de poder realizar un cobro justo a los contribuyentes, actualizando de manera eficiente la base de datos correspondiente. Para el sistema se toma en cuenta una referencia que permite obtener una base de bits, la cual se opera en la matriz de píxeles obtenida a partir del análisis de la imagen, después mediante un suavizado de la misma y la función Gaussiana se obtiene una imagen que permite determinar los bordes que componen una estructura o edificación, para posteriormente mediante una red neuronal analizar las líneas que conforman el edificio tomando en cuenta las variables más significativas. El problema en este tipo de sistemas es la determinación de las medidas y la toma valores erróneos durante la obtención de los espacios de estudio. Con la propuesta realizada se ha resuelto este problema a partir de imágenes satelitales y una red neuronal, generando de esta forma un cuadro de referencia encontrado a partir del estudio de la matriz de la imagen.

Palabras clave: análisis y procesamiento de imágenes, función Gaussiana, detección de bordes, redes neuronales, impuesto predial.

Design and Implementation of a System to Determine the Property Tax through the Processing and Analysis of Satellite Images

Abstract. This work presents the way to implement resources for analysis and image processing to obtain a registry of the properties not updated and registered with the public registry of the property and commerce, in order to be able to make a fair payment to the taxpayers, updating of Efficiently the corresponding database. For the system a reference is taken into account that allows to obtain a bit base, which is operated in the matrix of pixels obtained from the analysis of the image, then by smoothing it and the Gaussian function an image is obtained which allows to determine the edges that make up a structure or building, and then through a neural network analyze the lines that make up the building taking

into account the most significant variables. The problem in this type of systems is the determination of the measures and takes wrong values during the obtaining of the study spaces. With the proposal made, this problem has been solved from satellite images and a neural network, thus generating a reference chart found from the study of the image matrix.

Keywords: image analysis and processing, Gaussian function, edge detection, neural networks, property tax.

1. Introducción

Uno de los principales objetivos al implementar metaheurísticas en problemas de ingeniería, es el de resolver situaciones complejas y buscar soluciones factibles dentro de un intervalo definido por las cotas de diseño. Con apoyo de técnicas heurísticas como las redes neuronales, se logró encontrar las secciones que permiten obtener las características de interés para llevar a cabo el estudio de las regiones importantes de una imagen, el análisis y procesamiento digital de imágenes permite suavizar el archivo a trabajar y seccionar el área de análisis en regiones definidas como filas y columnas, lo que da como resultado una matriz de píxeles, realizando de esta manera la medición de las coordenadas del inicio y término de la región de análisis, tomándolo como punto de partida para la creación de un cuadro de referencias a examinar. Una vez llenado este requerimiento se puede regresar a la imagen suavizada con la que se determinará por medio de la función Gaussiana los bordes altos de la imagen, encontrando así los bordes generados para las estructuras de interés.

2. Antecedentes y trabajos previos

En [1] se diseñó un prototipo de sistema que usa distintos tipos de técnicas de computación para realizar análisis de imágenes mediante modelos de redes neuronales, dicho sistema usa información numérica (a nivel de píxel) para realizar tareas de interpretación de imágenes, identificando la presencia de ciertas estructuras oceánicas en África e Islas Canarias.

Mientras que en [2], se desarrolla una herramienta computacional para el apoyo médico que identifique si se tiene un tumor maligno o benigno, haciendo uso del procesamiento digital de imágenes, así como de las redes neuronales artificiales.

Por otra parte, en [3] se lleva a cabo el diseño de una herramienta geoespacial que tiene la capacidad de detectar automáticamente objetos en fotografías satelitales usando técnicas de procesamiento de imagen y visión artificial.

Además, en [4] se diseñó y construyó un sistema multibiométrico para la identificación de personal mediante visión artificial de iris y huella digital, la imagen obtenida se manipula para extraer información a través de la transformada de Wavelet de Haar y/o bifurcaciones y correlación adaptiva en 2D, para su clasificación se usan redes neuronales.

De igual manera, en [5] se intenta relevar las superficies boscosas utilizando procesamiento de imágenes satelitales para su posterior clasificación y ordenamiento de los recursos forestales en tierra del fuego.

Así mismo, en [6] se lleva a cabo el análisis y detección de edificios de forma automática/semiautomática sobre imágenes aéreas y satelitales mediante la obtención de umbrales haciendo uso de herramientas de cómputo.

También, en [7] se trata el diseño de un sistema automático de inspección que puede mejorar este proceso de control de calidad en las empresas textiles, empleando análisis de imágenes para eliminar las imperfecciones de las telas.

Además, en [8] se realiza la elaboración de un software de procesamiento de imágenes para los análisis metalográficos para la determinación y automatización del estimado métrico del grano y fases de ferrítico-perlíticas en aceros de bajo contenido de carbono.

Para finalizar, en [9] se llevó a cabo el desarrollo de una metodología para clasificación digital enfocada a la generación de cartografía referente a la Reserva Nacional Valdivia empleando imágenes satelitales, realizado mediante un método de clasificación supervisado y aplicando un algoritmo de máxima probabilidad.

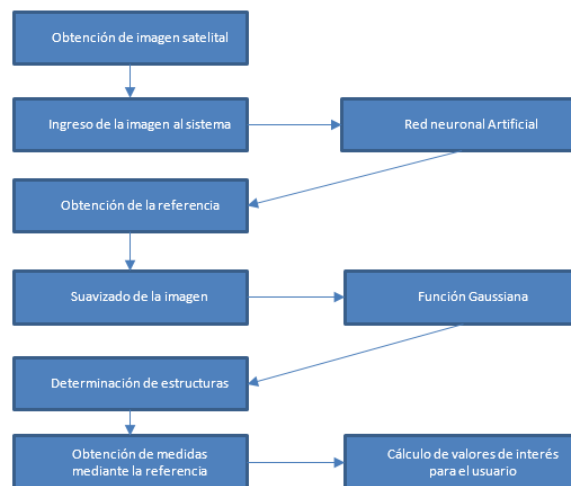


Fig. 1. Proceso considerado para el diseño del sistema

Respecto a los antecedentes mencionados, las desventajas son que los resultados obtenidos son muy ideales y difíciles de llevar a la práctica, sin embargo, esta investigación pretende contribuir a subsanar esta situación, mediante la implementación de restricciones que permitan obtener resultados factibles y que conlleven a utilizar esta metodología para diversos diseños con procesamiento de imágenes y redes neuronales en ingeniería.

3. Metodología

De acuerdo con el problema planteado, es posible adaptar el uso de las heurísticas para encontrar las regiones necesarias en el sistema, ya que son métodos de optimización y análisis que es posible utilizar para hallar soluciones a problemas no lineales o de complejidad analítica que son difíciles de especificar en un sistema no inteligente.

Basado en esto, se recurre al uso de las redes neuronales artificiales para el análisis de imágenes, con lo cual se utilizan dos tipos de entrenamientos para la red utilizada en este trabajo, cómo los son aprendizaje supervisado y aprendizaje por refuerzo.

Permitiendo así, entrenar a la red de tal manera que sea capaz de llevar a cabo el análisis requerido por el sistema y entrenándose así misma constantemente con cada uso. Para este caso se analiza el sistema de acuerdo con la figura 1.

3.1. Redes neuronales

Para la implementación de una red neuronal artificial es necesario entender que estas están basadas en una red neuronal biológica, por lo que su elemento básico es la neurona, por lo cual se busca generar una estructura que emule esta condición. Un sistema neuronal biológico este compuesto por millones de neuronas organizadas en capas. En la emulación de dicho sistema neuronal biológico, por medio de un sistema neuronal artificial, se puede establecer una estructura jerárquica similar a la existente en el cerebro.

Para llevar a cabo el diseño del sistema es necesario tener en cuenta el funcionamiento de una red neuronal artificial, por lo cual es importante analizar la estructura y arquitectura de las neuronas. Este trabajo se basa en una arquitectura de neurona de McCulloch-Pitts, esta funciona de manera muy similar a como funciona una neurona biológica. Para ser más precisos, es una unidad de cálculo que intenta emular el comportamiento de una neurona biológica, en la figura 2 se puede apreciar la arquitectura de la neurona artificial.

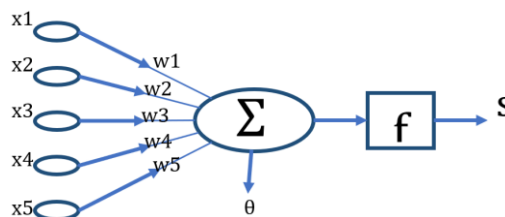


Fig. 2. Arquitectura de la neurona artificial.

Esta neurona artificial está formada por los siguientes elementos:

- X: Conjunto de n componentes denominados vector de entradas al sistema.
- W: Conjunto de pesos sinápticos de la conexión entre la entrada y la neurona.
- Σ : Sumatoria de procesamiento de los valores de entrada.
- θ : valor umbral.
- F: Función de activación que proporciona el estado de activación de la neurona en función.
- S: salida de la neurona.

Dentro de la neurona se lleva a cabo el cálculo de la siguiente ecuación (1):

$$\sum_{i=1}^n w_i x_i. \quad (1)$$

Debiendo tomar en cuenta que la arquitectura de McCulloch-Pitts permite dos posibles resultados 1 y 0, haciendo así necesario el comparar el resultado obtenido con el umbral de la neurona con el fin de detonar la función de activación (ecuación 2) enviando por medio del axón el resultado de la red neuronal:

$$s_1 \begin{cases} 1 & \text{si } \sum_{i=1}^n w_i x_i \geq \theta \\ 0 & \text{si } \sum_{i=1}^n w_i x_i < \theta. \end{cases} \quad (2)$$

Las neuronas funcionan como procesadores de información sencillos. De manera simplista las dendritas constituyen el canal de entrada de la información, el soma es el órgano de cómputo, el axón corresponde al canal de salida, y a la vez envía información a otras neuronas. Cada neurona recibe información de aproximadamente 10000 neuronas y envía impulsos a cientos de ellas.

3.2 Obtención de la referencia

En esta parte del sistema se buscará la imagen satelital, que se utilizará como referencia, en este caso para el análisis y como prueba del sistema. En esta sección se toma como referencia al Tecnológico de Estudios Superiores de Ecatepec, en específico el edificio de la división de informática como ejemplo, como se muestra en las figuras 3 y 4 respectivamente. Una vez obtenida la imagen es posible notar que en la parte inferior derecha contiene una medida de referencia aproximada, que se tomará como referencia en el sistema para aproximar una unidad de medida (véase figura 5).



Fig. 3. Plataforma Google Maps.



Fig. 4. Imagen satelital de prueba.

Tomando como ejemplo la imagen obtenida, esta es ingresada por medio del asistente de exploración de archivos del sistema, donde mediante el procesamiento de esta imagen es posible detectar la posición de la referencia que se tomará como parte de los datos necesarios para el cálculo de resultados.



Fig. 5. Referencia de medidas a tomar por el sistema.

Una vez obtenida la referencia, el sistema almacena el número de píxeles obtenidos de la aproximación métrica, con lo que genera una referencia visual que utiliza para una vez detectados los objetos sea posible comparar las dimensiones de cada objeto con el valor, generando una escala entre píxel por metro o número de píxeles en un metro de la misma manera en que se obtiene la referencia de la imagen satelital, en este paso por medio del procesamiento de imágenes se agrega dicha información. El resultado será algo similar a lo que se muestra la figura 6, en donde se puede ver el archivo en la etapa de procesamiento, parte en que se agrega el referente visual para el análisis de las dimensiones de las estructuras detectadas en la fotografía.

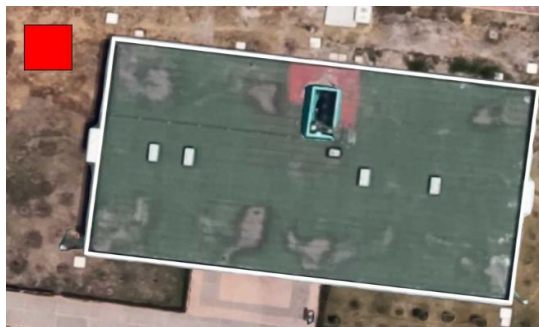


Fig. 6. Imagen satelital con referencia de medida.

La figura anterior ayuda a entender la importancia de la referencia de medida, ya que el sistema genera la referencia a través del estimado de medida entregado por la plataforma de donde se obtuvo la imagen. Basado en esto se obtendrán las unidades de medición requeridas para el sistema.

3.3 Detección de bordes

La detección de bordes es una técnica de análisis y procesamiento de imágenes que permite identificar objetos y aislarlos. Una vez obtenidas las salientes que conforman los bordes de una figura, es posible llevar un conteo de los objetos que se muestran en la imagen que se está utilizando como objeto de estudio, con el fin de almacenar mediante matrices las posiciones en coordenadas de cada punto encontrado como un borde.

En este caso se está haciendo uso de la llamada visión artificial, la cual es un subcampo de las matemáticas que abarca muchas técnicas que permiten llevar a cabo el tratamiento digital de imágenes. Con esta técnica se puede contar objetos específicos

haciendo uso de la herramienta OpenCV. Durante el análisis de imágenes es preferible que exista un contraste claro entre los objetos que se están buscando y el fondo. Esto permitirá al algoritmo realizar el análisis de una manera más eficiente.

El proceso se divide en 5 fases:

1. Convertir la imagen a escala de grises
2. Filtrar la imagen para eliminar el ruido
3. Aplicar el detector de bordes
4. Buscar los contornos dentro de los bordes detectados
5. Dibujar dichos contornos

Durante la aplicación de estos pasos se obtiene una imagen cada vez más procesada, dando como resultado un conjunto de matrices que contienen las posiciones en coordenadas de cada punto encontrado como un borde. Este proceso está siendo aplicado por medio de OpenCV haciendo uso del lenguaje Python.

4. Resultados y discusión

El conjunto encontrado se pinta sobre la imagen original, lo que permitirá como usuarios ver los bordes detectados por el sistema representados por líneas de color rojo sobre la imagen original, se usa este color con el objetivo de que sean lo más visibles posible y que resalten tanto para el sistema como para el usuario, y con lo que al finalizar, se puede visualizar los resultados en píxeles y la cantidad de objetos que se detectaron dentro de la imagen, con lo que se genera el cálculo de las medidas en metros, como se muestra en la figura 7.

```
He encontrado 72 objetos
Perimetro = 1783
Lado = 445.75
Area = 166282.5
```

Fig. 7. Medidas en píxeles y conteo de objetos.

Para finalizar, como resultado se muestran las medidas obtenidas en metros sobrepuestas a la imagen como se observa en la figura 8, que permitirá realizar los cálculos correspondientes para encontrar los valores requeridos.



Fig. 8. Medición de objetos.

5. Conclusiones y trabajo futuro

Se ha logrado desarrollar el diseño de un sistema para la determinación del impuesto predial a través del procesamiento y análisis de imágenes satelitales, utilizando métodos como las redes neuronales artificiales, análisis y procesamiento de imágenes, determinación de medidas con respecto a matrices generadas a partir de imágenes. Es posible destacar que el uso de las heurísticas en la implementación del sistema diseñado ha sido de mucha utilidad debido a la complejidad de la detección de ciertos componentes requeridos por el sistema, dentro tanto de las imágenes como de las matrices de bits utilizadas durante la detección de estructuras.

Con el fin de continuar con el avance en el desarrollo alcanzado dentro de este trabajo, se propone implementar otros algoritmos de *Deep learning* para la actualización automática de valores, que permitan al sistema trabajar más inteligentemente, para de esta manera lograr realizar el objetivo con mayor eficiencia y así el usuario pueda hacer uso de los resultados con base a información precisa.

Agradecimientos. Los autores agradecen al Tecnológico de Estudios Superiores de Ecatepec, en específico a la división de informática por el apoyo brindado.

Referencias

1. Torres, A.J., Cantón, M.: Detección y reconocimiento de estructuras oceánicas en imágenes AVHRR en el área de las islas canarias y NO de África mediante técnicas de computación conexionista. *Revista de Teledetección*, vol. 13, pp. 5–12 (2000)
2. Flores Gutiérrez, H.: *Redes Neuronales Artificiales aplicadas a la detección de Cáncer de Mama*. Tesis doctoral, Instituto Politécnico Nacional, México (2015)
3. Iglesias García, A.: *Desarrollo de una herramienta para la detección de objetos de interés en datos multimedia georreferenciados*. Tesis doctoral (2018)
4. Collado, L.: Los bosques de Tierra del Fuego. Análisis de su estratificación mediante imágenes satelitales para el. *Multequina*, n° 10, pp. 1–15 (2001)
5. Cristina, F., Dapoto, S.H., Artola, V., Russo, C.C., Viegas, G.M.: Detección de edificios en imágenes urbanas. In: *XV Congreso Argentino de Ciencias de la Computación* (2009)
6. Alvarado, J.D., Alvarado F.J.: Análisis de textura en imágenes a escala de grises, utilizando patrones locales binarios (LBP). *ENGI Revista Electrónica de la Facultad de Ingeniería*, Vol. 1, No. 1 (2015).
7. Llulluna, L., Fredy, R.: Procesamiento de imágenes mediante software libre python para el análisis metalográfico en aceros de bajo contenido de carbono. Quito, pp. 60–79 (2014)
8. Segura, R., Trincado, G.: Cartografía digital de la Reserva Nacional Valdivia a partir de imágenes satelitales Landsat TM. *Bosque (Valdivia)*, vol. 24, n° 2, pp. 43–52 (2003)

Embedded System for Human Detection Applied to Domotics

Oscar Arturo González González¹, Alina Mariana Pérez Soberanes¹,
Víctor Hugo García Ortega¹, Julio César Sosa Savedra²

¹ Instituto Politécnico Nacional, ESCOM, Ciudad de México,
México

² Instituto Politécnico Nacional, CICATA, Querétaro,
México

oscar.ar-56@hotmail.com, psoberanes.mariana@hotmail.com
vgarciao@ipn.mx, jcsosa@ipn.mx

Abstract. This paper presents the development of an embedded system prototype that performs the home security monitoring, through image processing and classification algorithms to detect a human presence. If a human presence is detected, the system will send an alert message to the user. The embedded system is implemented on a Raspberry Pi 3 B, supported by a Pyroelectric Infrared Radial (PIR) motion sensor and a Raspberry Pi Camera V2. The algorithms are implemented in C language and were designed to take advantage of the hardware resources of the platform, through High Performance Computing (HPC) techniques. The selected classifier is a multilayer perceptron. This classifier obtained an accuracy of 96 %, approximately.

Keywords: image processing, domotics, embedded system, multilayer perceptron, HPC, Raspberry Pi.

1 Introduction

Statistics from the nonprofit organization *Observatorio Nacional Ciudadano* in Mexico indicates that only in the period March - April 2019, there was an increase of burglary investigation processes from 6,734 to 7,017 [1], that is an increase of 4.20%, in the number of investigation processes from March to April 2019. The situation of burglary in Mexico generates the need for people to keep their home safe. A possible solution to this problem is to develop new technologies that allow home automation. These technologies focus on the automation of processes within a building. This improves the quality of life and comfort, inside the home. It also allows constant monitoring of the place, through the Internet and the use of alarms, sensors or surveillance cameras. This domotics development has driven the creation of security systems such as those that are installed and monitored by private companies, which usually have a monthly cost per service, or commercial devices that can be installed and configured by the user. Some of these alternatives tend to be inefficient, and some others are often expensive or cannot be obtained by the majority of the population.

There are some papers proposed related to the home security monitoring and human detection. All these applications are developed in a Raspberry Pi platform and are supported by a camera and by sensors. The first paper *A New IoT Combined Body Detection of People by Using Computer Vision for Security Application* [2] presents a system developed on the Raspberry Pi platform with sensor elements: Raspberry Pi camera and a Pyroelectric Infrared Radial (PIR) sensor, which is used for motion detection. Once the image is obtained, the algorithm of the Histogram of Oriented Gradients (HOG) extracts the object's features and a Support Vector Machine (SVM) is used for the classification and learning of the human form. If a human presence is detected, the image is sent to a smartphone over the Internet, using the Telegram application. The artificial vision tool used for the development of this system is OpenCV. Regarding the results of the proposed method, it is presented that it was obtained a detection rate of $93.89 \pm 5.3\%$.

The paper *Human Detector and Counter Using Raspberry Pi Microcontroller* [3] presents a system that identifies humans entering a room through a door. The system performs digital image processing using HOG as a feature extraction algorithm and a SVM for the classification, based on the OpenCV tool and developed in the Python programming language. It is specified that it is supported on a Raspberry Pi and uses a Raspberry Pi Camera, other sensors are a pair of PIR sensors, to indicate to the system the capture of input or output images; and a Bluetooth module (HC-05) to send the required data to another system through serial communication. It has an efficiency of 83 %. In addition, it is mentioned that the use of Raspberry Pi platform avoided the use of desktop CPU and as a result it was reduced the system power consumption and the cost.

In this paper it is proposed an embedded system for home security monitoring, which is implemented in a Raspberry Pi 3 B platform using HPC techniques, thereby taking advantage of the hardware and software resources of the platform. The system detects the human presence and sends an alert message to the user. This allows users to keep their home safe through the remote monitoring. The system uses a motion sensor to perform the motion detection. The images are obtained from Raspberry Pi Camera V2. Also the system implements image processing and classification algorithms. This feature allows the system to improve its robustness and to perform a reliable detection of human presence inside the enclosure. Additionally, the system is considered to be scalable as a security module of a robust home automation system. This system can include the control of doors, windows, temperature, among other possible modules. The general diagram of the proposed system is presented in Fig. 1.

2 Methodology

Raspberry Pi is a suitable platform for home automation systems. This is the selected platform for the implementation of the system. In [4] is mentioned that Raspberry Pi is a platform that supports a large number of input and output peripherals. Raspberry Pi is also suitable for interacting with different devices,

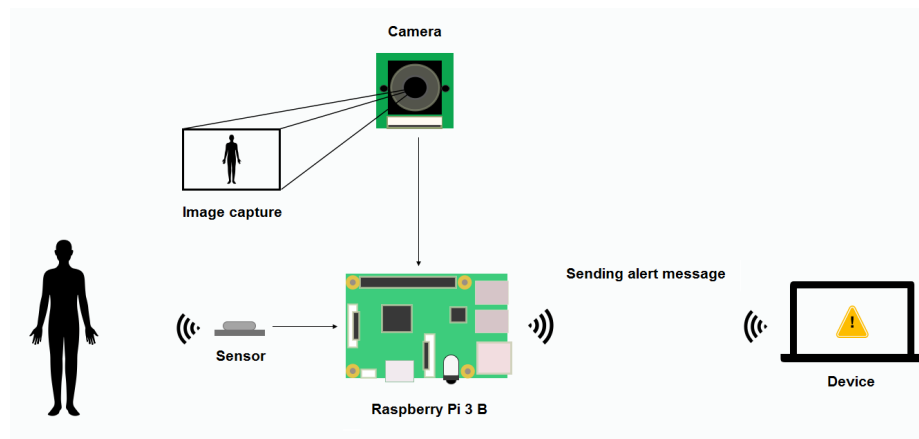


Fig. 1. Proposed system.

allowing its use in a wide range of applications. On the other hand, the ability to access the Internet makes the Raspberry Pi suitable for home automation related applications.

The system is composed of four main modules:

1. **Motion detection module:** In this module the embedded system through the AS312 PIR sensor detects movement and the Raspberry Pi Camera Module V2 perform the acquisition of images.
2. **Image processing module:** In this module the image subtraction for motion detection is performed. Then the algorithms for the image segmentation and the image feature extraction are performed.
3. **Classification module:** In this module the descriptor vectors obtained from the image feature extraction are analysed by the classification algorithm.
4. **Communication module:** This module sends a notification to the user's device to notify the possible human presence.

Each of the previous modules will be described in the following sections.

2.1 Motion Detection Module

For the motion detection it was implemented a circuit for the AS312 sensor, which is powered with 3.3 V. The circuit output depends on the activation of the AS312 sensor, if the sensor detects movement the output will be high (3.3 V), otherwise if the sensor does not detect movement, the output will be low (0 V). The sensor has a detection distance of 4 meters and a viewing angle of 60° approximately. The reading of the circuit output is performed through the General Purpose Terminals (GPIO) of the Raspberry Pi. This reading allows to send the signal to the system indicating if movement was detected or not.

This signal is read through the WiringPi library, which is a GPIO access library written in C language for the BCM2835, BCM2836 and BCM2837 System on Chip (SoC) devices used in all Raspberry Pi platforms [6]. If movement is detected the system will capture an image that will be processed by the next system module. This image capture is performed through the Raspberry Pi Camera Module V2, which is connected to the Raspberry Pi platform through the Camera Serial Interface (CSI) and is controlled by the *raspistill* commands of the operating system of the Raspberry Pi (Raspbian Stretch).

2.2 Image Processing Module

This module is divided into 4 submodules:

A. Image acquisition, grayscale conversion and noise filter application.

The image obtained by the Raspberry Pi camera V2 is in BMP format -Bitmap-, which is a bitmap file, these kind of files are uncompressed and have 24 bits per pixel, also this kind of image do not have any type of processing of data, or alteration, so the original data is preserved [5]. Once the BMP image is obtained a grayscale conversion, and then a Gaussian filter is applied to the image. This filter allows to attenuate the noise and has a better frequency response. For its implementation, a convolution operation is performed with a mask, with a window size of 3x3 (2). Equation (1) is used to obtain the mask and the value of σ is 1.0:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

$$G = \begin{pmatrix} 0.0751 & 0.1238 & 0.0751 \\ 0.1238 & 0.2042 & 0.1238 \\ 0.0751 & 0.1238 & 0.0751 \end{pmatrix}. \quad (2)$$

To avoid using the floating point unit by the Raspberry Pi processor, an approximation of (2) is obtained in (3):

$$G = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \quad (3)$$

This mask or kernel is multiplied by each pixel and its neighbors, in order to homogenize the colors of nearby neighbors. This achieve the elimination of white dots or white noise in an image, because the noise usually takes different pixel values from those in a neighborhood of pixels.

The Broadcom BCM2837 processor of the Raspberry Pi 3 B card has four cores, so the processing can be divided into four processing blocks. The Gaussian filter is implemented by dividing the processing in parallel into four processes

working on the same image (data parallelism). This allows to take advantage of the resources provided by the platform. The implementation of this Gaussian filter and all the algorithms that will be presented in the next sections are built in C language, which is a programming language that allows the compiler to generate the assembly code directly from the processor.

B. Motion detection in an image. Once the processing of the previous modules was applied to a base image - which is captured at the first process of the execution of the system -, this image is set as a background. Then if motion is detected by the sensor movement a subsequent image will be captured. The motion detection in the image is performed through implementing the background subtraction method, which is performed with the subtraction operation between the two images, with this operation is obtained the pixel differences between two images and thus be able to identify if there are differences between one scenario and another. Mathematically, the subtraction can be represented as $f = x - y$, where x and y are images respectively.

The simplest case to perform the subtraction and not to obtain negative values is to apply the absolute value function (4):

$$f_a = |x-y| . \quad (4)$$

Considering that the subtraction has the property of being commutative. The result of the subtraction between the two images must be a positive value. For this the value obtained is verified, if it is negative, the result of the subtraction will be replaced by the value of 0, otherwise the value obtained from the subtraction is assigned.

C. Image segmentation. This submodule is dedicated to the subdivision and analysis of image regions for segmentation. The region corresponding to the object of interest is represented with a pixel value of 255, while the region belonging to the background of the image adopts a pixel value of 0. First, the Otsu segmentation algorithm is performed to obtain an automatic global threshold depending on the analysis of the image pixels, it requires three parameters: the image obtained from the motion detection module, its width and height. With this operation the threshold is obtained and then it will be used in the image thresholding. The Otsu method requires the values of the image histogram, and are used to perform the consequent operations to get the maximum value. The Otsu method allows segmentation of an image through discriminant analysis, separating the image into two classes C_0 and C_1 this classes correspond to object and background. This method automatically finds the threshold of an image using its gray segmentation levels [7]. Class values take values $C_0 = \{0, 1, \dots, u\}$ where u is the gray level and $C_1 = \{u + 1, u + 2, \dots, L - 1\}$ where L is the number of gray levels. To obtain the probability of occurrence of a gray level P_i in a digital image the following expression is used, where N is the number of pixels in gray level of the image and f_i is the number of frequency of pixels with the

same gray level (5):

$$P_i = \frac{f_i}{N} . \quad (5)$$

The classes C_0 and C_1 are given by (6) and (7):

$$C_0 : \frac{P_1}{\omega_0(u)}, \dots, \frac{P_u}{\omega_0(u)} , \quad (6)$$

$$C_1 : \frac{P_{u+1}}{\omega_1(u)}, \dots, \frac{P_L}{\omega_1(u)} . \quad (7)$$

Equations (8) and (9) define ω_0 and ω_1 :

$$\omega_0(u) = \sum_{i=u+1}^L P_i , \quad (8)$$

$$\omega_1(u) = \sum_{i=u+1}^L P_i . \quad (9)$$

Subsequently, the average for each class is obtained, given by μ_0 and μ_1 (10), (11).

$$\mu_0 = \sum_{i=1}^u \frac{i * P_i}{\omega_0(u)} , \quad (10)$$

$$\mu_1 = \sum_{i=u+1}^L \frac{i * P_i}{\omega_1(u)} . \quad (11)$$

Once the means of the classes are obtained, the variance is obtained (12):

$$\sigma_B^2 = \omega_0 * (\mu_1 - \mu_T)^2 + \omega_1 * (\mu_2 - \mu_T)^2 , \quad (12)$$

where μ_T is the average obtained from the total pixels of the image in gray levels. The threshold t^* is selected by maximising the separation of classes from the image and the coefficient between the variance between classes is maximised. The greater the variance between classes and the smaller the variance within them, the discrimination will be better between the groups (13):

$$t^* = MAX(\sigma_B^2(u)) \quad 1 \leq u \leq L . \quad (13)$$

Subsequently, the thresholding of the image is performed using the threshold value obtained with the Otsu method. To do this, we proceed to verify if the value of the pixel of the image to be thresholded is greater or less than the desired threshold and depending on this, modify its value in 0 or 255 (14):

$$f(i, j) = \begin{cases} 255 & \text{si } l > u, \\ 0 & \text{si } l < u. \end{cases} \quad (14)$$

To perform the image threshold, it is required to define a threshold value u . If the gray level value l of an image $f(i, j)$ is greater than the threshold u .

After the Otsu threshold, the image is partitioned into subimages of specific size. In this subimages the pixel values are ordered and if the neighborhood is adequate, the area is dilated. The pixel values of the subimage are saved, considering the binarized image obtained with the Otsu method and once the values are sorted, a value of 0 or 255 is assigned to the analyzed pixel depending on the relationship that it has with the neighbor pixels, related with the pixel average. This processing of the white pixels allows to obtain a better image of the section that belongs to the object. This operations require further processing, thus operations are divided by four processes to take advantage of the hardware and software resources.

Finally, the Sauvola method is used to complete thicken regions in the image in addition to correcting shadows that are not related to the object. This method is used to reduce light intensities in an image. This method uses the average intensity of the values of a given region of pixels and their standard deviation. This region is defined within a $b \times b$ neighborhood or matrix, with a k parameter which varies between 0.2 and 0.5. This is implemented in the equation (15) obtained from [8]:

$$T(x, y) = m(x, y) \left[1 + k \left(\frac{\sigma(x, y)}{R} - 1 \right) \right] , \quad (15)$$

where:

- $\sigma(x, y)$: Is the standard deviation of the neighborhood.
- R : Is maximum value of the standard deviation, $R = 128$ with the maximum intensity being 255 in a neighborhood with $b = 15$.
- k : A value between 0.2 and 0.5.
- $m(x, y)$: Arithmetic mean of the neighborhood.
- $T(x, y)$: Value of the threshold in the neighborhood.

The processing result images of image subtraction and the image segmentation submodule are presented in the following Fig. 2.

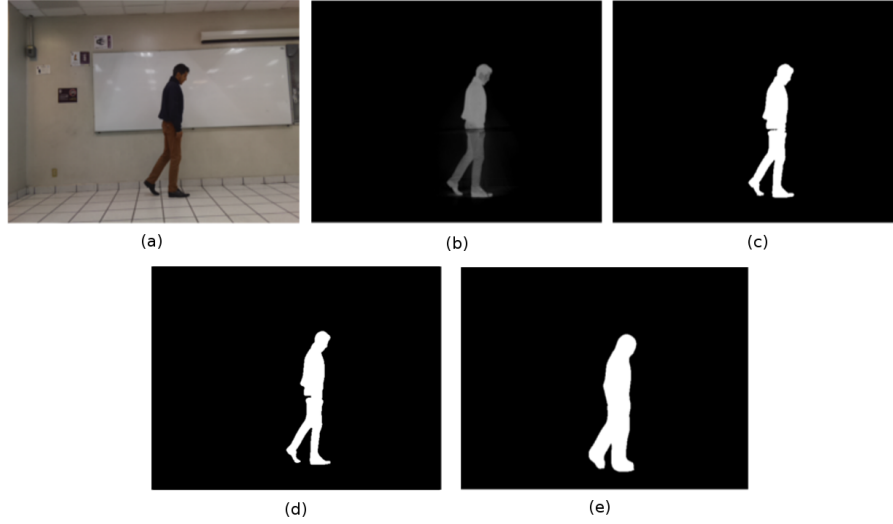


Fig. 2. Original image (a), subtraction image (b), Otsu image (c), local threshold image (d), Sauvola image (e).

D. Feature extraction. This submodule obtains the descriptors of the segmented object. The invariant descriptors selected for this submodule are the Hu moments which are descriptors that extract invariant features to translations, rotations, changes of scale and illumination. These invariant moments are obtained from the geometric moments [7].

For the discrete case of an image $f(i, j)$ the standard geometric moments are obtained in (16):

$$M_{pq} = \sum_i \sum_j i^p j^q f(i, j) \quad \forall p, q = 0, 1, 2, \dots \quad (16)$$

To obtain the central moments of the object (17):

$$\mu_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q f(i, j) \quad \forall p, q = 0, 1, \dots \quad (17)$$

where \bar{x} and \bar{y} correspond to the coordinates of the center of gravity of the analysed object, which are obtained from the geometric moments of order 10, 01 and 00 (18):

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (18)$$

To obtain the invariants to changes of scale, each moment must be divided by a normalization factor (19) that cancels the scaling effect [7]:

$$\gamma = \frac{p+q}{2} + 1 \quad \text{for } p+q = 2, 3, \dots \quad (19)$$

Equation (20) is used to calculate the normalized central moments:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}} . \quad (20)$$

By obtaining the normalized central moments with (20), which allows to obtain the seven invariant moments –Hu moments– (21):

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ &\quad (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] . \end{aligned} \quad (21)$$

2.3 Classification Module

Once the moments of Hu have been obtained from (21), they must be processed by the classification module to determine if the shape obtained from the image corresponds to a human form. For this, a classifier is required, in this case the multilayer perceptron was chosen, because it allows the classification of linearly non-separable patterns and it is adequate to solve biclass classification problems [7]. The *backpropagation* algorithm is used for multilayer perceptron training. The training of the network requires training patterns. This patterns correspond to a group of feature vectors obtained from segmented objects of a set of training images, which were processed with the algorithms of the previous modules. The number of elements of the training set is 182, divided into 91 elements that correspond to human form and the remaining 91 are non-human forms.

The 91 human forms were obtained from 6 volunteers, each person was photographed in different positions resembling the walking position of a person, see Fig. 3. On the other hand, the 91 images of non-human form are images of ordinary objects that are commonly found in a house, such as pets, balls, lamps.

The final architecture of the neural network, after performing several tests, is presented in Fig. 4, which shows an architecture of 7 inputs, 5 neurons were required in the hidden layer and only one neuron in the output layer. The 7 inputs correspond to the 7 elements of the feature vector obtained from Hu moments (21), these are the image descriptors. The activation function of the hidden layer neurons is the *hyperbolic tangent*, which allowed the backpropagation algorithm to train faster. For the output layer it is assigned the *sigmoid function*, because



Fig. 3. Sample of human form images obtained from the set of 91 training elements.

of the output of the network is between 0 and 1 (1 correspond to human form and 0 correspond to non-human form).

Before training, the feature vectors of the set of 182 images were normalized with (22) obtained from [9]:

$$x'_j = \frac{1 - \exp\left(-\frac{x_j - \mu_j}{\sigma_j}\right)}{1 + \exp\left(-\frac{x_j - \mu_j}{\sigma_j}\right)}, \quad (22)$$

where x'_j corresponds to the normalized value, x_j is the value to be normalized, μ_j is the average of the corresponding dimension and σ_j is the standard deviation of the corresponding dimension. This normalization allows to obtain a set of normalized data within the range of -1 to $+1$, which is the range of values that allow the activation function *hyperbolic tangent* to obtain appropriate activation values and to avoid values that can saturate the function.

Once the values of the normalized dataset are obtained these are labeled with the value of 0 and 1 (human form: 1 and non-human form: 0). This dataset is processed by the *backpropagation* training algorithm. The training parameters were an $\alpha = 0.5$ and a $\eta = 0.25$. The algorithm converges when the error between the target and the multilayer perceptron output is as small as possible, updating in each iteration the set of weights and bias of each layer connection.

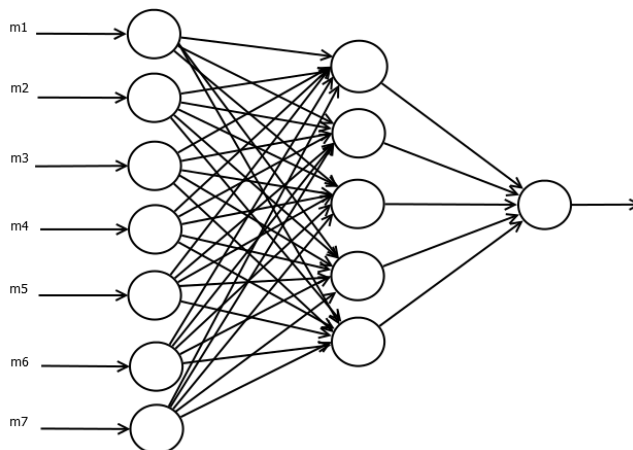


Fig. 4. Final architecture used for the classification module.

2.4 Communication Module

The communication is a simple client-server architecture, in which the image processing and classification server is implemented on the Raspberry Pi platform. The communication is implemented using a socket. The client is implemented on a computer with a Linux distribution, such as Ubuntu, which receives the status of the home monitoring performed by the server, through port 5000. The server and the client are developed under the TCP/IP protocol stack in C language.

3 Results

The performance of the classification model can be analyzed in a confusion matrix in Table 1. This matrix is a table of $N \times N$ –where N is the number of classes– that summarizes the level of success of the predictions of a classification model. In the case of the multilayer perceptron designed and implemented, the number of classes is equal to 2, because of the problem of recognizing a human form, in this case, is a binary classification problem.

The confusion matrix obtained is a matrix of 2×2 . And the results were obtained using a test set of 46 elements. The dataset of 228 images is divided in 182 training set images and 46 test set images.

Table 1. Confusion matrix of the classification model.

	Human form	No human form
Human form	22	1
No human form	1	22

The confusion matrix in Table 1 describes that the 23 images containing human form were correctly classified 22 and incorrectly classified 1 image. And the other 23 images that do not contain human form were correctly classified 22 and 1 incorrectly classified. Of the total of the 46 elements of the test set, 95.65% were classified by the model correctly, while 4.34% were incorrectly classified.

The execution time in which the Raspberry Pi platform performs the image processing, from the image acquisition module to the classifier module, is approximately of 21 seconds per image. This was measured considering the execution of the algorithms sequentially. Because of these results it was decided to apply HPC techniques, based in data parallelism. These techniques improved the performance, obtaining only 7 seconds of image processing, approximately.

The Raspberry Pi Camera V2 obtains an image in an amount of time of approximately 1 second, this execution time was obtained due to the reduction of the configuration time of the camera to only 500ms, which is the minimum required for the camera. As a result the total execution time is approximately of 7 to 8 seconds per image. Also the PIR sensor reading is configured to perform a sampling time of 0.5 seconds.

4 Conclusions

The development of the system on a SoC-based platform and the use of HPC techniques allowed to obtain a better computational performance. This allowed to take advantage of the available software and hardware resources of the Raspberry Pi 3 B. Also the implementation in C language, contributed to obtain a better performance.

About the image processing algorithms, the lighting changes on the test scenario generated noise in the sample images. To avoid this it was necessary to implement global and local thresholds in the image regions, in addition to the Sauvola algorithm. This algorithms of the segmentation module obtained an image without white noise or lighting changes, this result improved the robustness of the system, because of the avoiding of inaccuracies. Also obtaining the characteristic features of an object in an image through the Hu moments allowed to avoid the problem of translations, rotations, changes of scale and illumination.

The classification module required the soft-max normalization algorithm to improve the separation of the descriptor vectors of each object detected in an image. This normalization also improved the execution of the backpropagation algorithm. The result of the classifier were satisfactory. It was obtained an accuracy of approximately 96%. The result allows the system to be placed as a reliable and robust application for home automation. In comparison with the development presented by Othman and Aydin in [2], it has an accuracy of 93%, approximately. And Mathur et al. in [3] presents a development with an accuracy of only 83%. There is a significant accuracy difference and the development presented in this paper is implemented taking advantage of the hardware and software resources of the Raspberry Pi platform, through HPC techniques.

Acknowledgements. The authors thank the support of Instituto Politécnico Nacional (IPN). SIP-IPN contributed to the development of this work through 20180341 project.

References

1. Observatorio Nacional Ciudadano: Reporte sobre delitos de alto impacto. Apr. 2019. [Online]. Available: <http://onc.org.mx/wp-content/uploads/2019/06/mensual-abril.pdf>. [Accessed: Jun. 11, 2019]
2. Othman, N.A., Aydin, I.: A new IoT combined body detection of people by using computer vision for security application. In: 9th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 108–112 (2017)
3. Mathur, S., Subramanian, B., Jain, S., Choudhary, K., Prabha, D.R.: Human detector and counter using Raspberry Pi microcontroller. In: 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1–7 (2017)
4. Maksimovic, M., Vujovic, V., Davidovi, N., Milosevic, V., Perisic, B.: Raspberry Pi as Internet of Things hardware: Performances and Constraints. In: IcETran2014, Vrnjacka Banja, Serbia (2014)
5. Villagómez, C.: El formato BMP. CCM (2017). [Online]. Available: <https://es.ccm.net/contents/719-el-formato-bmp>. [Accessed: Jan. 21, 2019]
6. Wiring Pi: GPIO Interface library for the Raspberry Pi. Wiring Pi. [Online]. Available: <http://wiringpi.com/>. [Accessed: Mar. 20, 2019]
7. Rodríguez, R., Sossa, J.H.: Procesamiento y análisis digital de imágenes. 1ra ed. México: Alfaomega Grupo Editor, pp. 180–230, 250–290, 297–350 (2012)
8. Cortés, J., Chaves, J., Mendoza, J.: Comparación cualitativa y cuantitativa de las técnicas básicas de umbralización local para el procesamiento digital de imágenes. *Scientia et Technica* 2(51), p236–241. [Online]. Available: <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/7561/4707>. [Accessed: Dec. 11, 2018]
9. Kevin, P., Priddy, L.: Artificial Neural Networks: An Introduction. Washington: SPIE, pp. 16–17 (2005)

Estudio de herramientas para alcanzar niveles de calidad de software a nivel de proceso

Yadira Jazmín Pérez Castillo, Sandra Dinora Orantes Jiménez

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México
yaddy9011@gmail.com, dinora@cic.ipn.mx

Resumen. El área de la ingeniería de software se ha encargado de proporcionar herramientas para mejorar los procesos de construcción de software y, por ende, la calidad en el mismo, por lo tanto, para lograr esto, debe guiarse de métodos, modelos y/o prácticas que permitan alcanzarla. Este trabajo de investigación intenta proveer una revisión de los instrumentos existentes en el mercado, que permitan a una empresa desarrolladora de software alcanzar niveles de madurez a nivel proceso o bien, contar con estrategias utilizadas para aumentar la calidad de los productos de software.

Palabras clave: ingeniería de software, calidad de productos de software, procesos, herramientas, madurez de procesos.

Study of Tools to Reach Software Quality Levels at the Process Level

Abstract. The area of software engineering has been responsible for providing tools to improve the processes of software construction and, therefore, the quality in it, therefore, to achieve this, it should be guided by methods, models and / or practices that allow to reach it. This research work tries to provide a review of the instruments in the market, which allow a software developer to reach maturity levels at the process level or to have strategies used to increase the quality of software products.

Keywords: software engineering, quality of software products, processes, tools, process maturity.

1. Introducción

La calidad de software y las contrariedades que lleva consigo conseguir un nivel de calidad aceptable, es un problema que tiene sus raíces desde la llamada crisis del software y para mitigar sus estragos, varias estrategias han surgido en los últimos años y han proveído técnicas que ayudan a una mejora de la calidad en los procesos de desarrollo de software. Según [1], la creciente preocupación por la calidad en la industria del software tiene como objetivo principal el desarrollo sistemático de productos y servicios de mejor calidad y el cumplimiento de las necesidades y

expectativas de los clientes. La calidad de software se puede representar de acuerdo con el enfoque de evaluación, ya sea a nivel de proceso, producto o calidad en uso [2].

Por lo tanto, el presente trabajo se centrará en el tipo de evaluación a nivel de proceso, tomando en consideración que así se asegura el desarrollo de un producto de software de calidad y por ende, que posee la característica de usabilidad; por lo anterior, el objetivo de este trabajo es comparar las herramientas existentes en el mercado actual, que permiten acceder a una guía para alcanzar un nivel de calidad aceptable en los procesos de desarrollo, es por ello, que resulta favorable conocer cuáles modelos o estándares poseen una visión integradora de la calidad del producto de software o cuáles resultan adecuados para el ámbito de una organización.

Este documento se encuentra distribuido de la siguiente manera: en la *sección 2*, se dan a conocer algunos modelos importantes a nivel de proceso para la calidad de software, en la *sección 3*, se proporciona una discusión que permite dar resultados de esta investigación a través de un cuadro comparativo con base en características principales de los modelos descritos en la *sección 2* y sus apartados. Finalmente, en la *sección 4*, se establecen las conclusiones del trabajo realizado y comentarios acerca de trabajos futuros.

2. Modelos a nivel de proceso para la calidad de software

Realizando el análisis de investigación de la bibliografía existente, correspondiente al tema a abordar en este trabajo, es posible encontrar algunos modelos actualmente reconocidos por la industria de la ingeniería de software, entre ellos, ISO/IEC 15504 (SPICE), BOOTSTRAP, DROMEY, IEEE / EIA 12207, COBIT (Control Objectives for Information and related Technology, Objetivos de Control para Información y Tecnologías Relacionadas), ISO 90003, ISO 20000, CMMI (Capability Maturity Model Integration, Modelos de Madurez de Capacidades Integrados), PSP (Personal Software Process, Proceso de Software Personal), TSP (Team Software Process, Equipo de Procesos de Software), cuyo objetivo en común es proveer una evaluación de la calidad del software a nivel proceso; dentro de los siguientes apartados, se realiza una breve descripción de cada uno de ellos, con el fin de conocerlos, resaltando sus principales características, objetivos y funciones.

2.1. ISO/IEC 15504 (SPICE)

Permite adaptar la evaluación para procesos en Pequeñas y Medianas Empresas (PYMES) y grupos de desarrollo pequeños, mediante la estructuración en 6 niveles de madurez: Nivel 0- Organización inmadura, Nivel 1- Organización básica, Nivel 2- Organización gestionada, Nivel 3- Organización establecida, Nivel 4- Organización predecible y Nivel 5- Organización optimizando. Su objetivo es llegar a que la organización logre ser madura, lo cual, conlleva que la organización tenga procesos definidos, responsabilidades definidas, predicción de resultados, productos entregados con calidad, que las entregas se den en los tiempos pactados, incrementar la productividad, clientes satisfechos, y empleados felices [2].

ISO/IEC 15004 (SPICE) trata de una herramienta con los siguientes objetivos determinados en [3]:

- Proponer y desarrollar un estándar de evaluación de procesos de software.
- Evaluar su desempeño mediante su experimentación en la industria emergente del desarrollo de software.
- Promover la transferencia de tecnología de la evaluación de procesos de software a la industria del software a nivel mundial.

2.2. BOOTSTRAP

Es una metodología de valoración de la calidad que permite la mejora de procesos a partir de 6 actividades básicas: examinar la necesidad, iniciar proceso de mejora, preparación y dirección de la evaluación, análisis de resultados, implantación y finalización de mejoras [2].

Bootstrap Institute es el instituto que dirige los destinos de la metodología y está constituido por: miembros fundadores, corporativos y demás integrantes; dentro de sus principios están: que la metodología sea accesible a todos y crezca de forma que permita mejoras, que la evolución de la metodología sea democrática, que provea un servicio a la industria y que opere como una empresa no lucrativa [4].

2.3. DROMEY

Es un modelo adaptable para evaluar varias etapas del proceso de desarrollo, tales como levantamiento de requisitos, diseño e implementación. Se estructura con características y sub-características de calidad; propone tres modelos distintos para cada etapa de construcción del producto: modelo de requerimientos, modelo de diseño y modelo de calidad de la implementación, a partir de la evaluación establecida en cinco etapas, para características como: eficiencia, confiabilidad, mantenibilidad, portabilidad, facilidad de uso y funcionalidad [2].

Es un modelo de calidad a medida, propuesto por Robert Geoff Dromey en 1995. Su propósito es el de trabajar con una estructura que permite construir y utilizar un modelo de calidad práctico para evaluar las etapas de determinación de los requerimientos, diseño e implementación. Esta información puede ser usada para elaborar, comparar y evaluar la calidad de los productos de software. Este modelo plantea la calidad del producto por medio de la definición de sub-características que pueden ser medidas y evaluadas como características. También, permite aumentar el entendimiento respecto de la relación entre los atributos (características) y los sub atributos de calidad [5].

2.4. IEEE / EIA 12207

Este estándar establece un marco de trabajo común para el ciclo de vida del desarrollo de software, a partir del planteamiento de procesos, actividades y tareas que pueden ser aplicadas durante la adquisición, suministro, desarrollo, operación, mantenimiento y/o despliegue de un producto software [2]. IEEE/EIA 12207: Procesos del Ciclo vital del software, se beneficia de una herencia que atraviesa un cuarto del siglo del trabajo de estándares de proceso del software, el estándar extensamente se considera como el suministro de una base para el comercio mundial en servicios del software. Al lado de la ISO 9.000, es probablemente el estándar más relacionado con la información importante con relación a la tecnología [6]. La Fig. 1, muestra la

estructura del modelo, como se puede observar, consta de 3 partes, las cuales permiten a una empresa ir incorporando el modelo progresivamente.

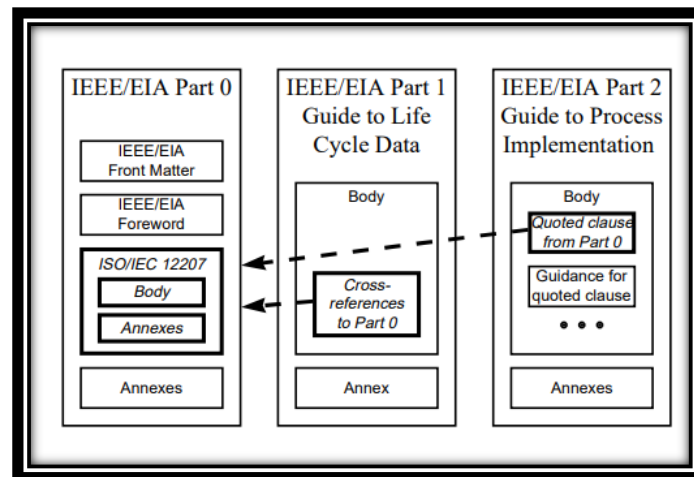


Fig. 1 Estructura IEEE / EIA 12207 [2].

2.5. COBIT

Se caracteriza por ser orientado a negocios y procesos, además de ser basado en controles, trabaja con siete criterios de información que son definidos como requerimientos de control del negocio: efectividad, eficiencia, confidencialidad, integridad, disponibilidad, cumplimiento y confiabilidad [2].

COBIT se ha puesto en práctica en muchos países desde su introducción en 1996. Uno de los factores que dan popularidad a COBIT es que su extenso resumen, marco, objetivos de control, guías de la dirección y juego de herramientas de realización son gratuitos. El pago sólo se requiere para las guías de auditoría. COBIT parece hacerse un marco influyente para el control y el gobierno de TI (Tecnologías de la Información) se certifican a por el significado y diversidad de las organizaciones en cual se ha utilizado. Además, COBIT está actualmente en su quinta edición y consta de 3 niveles de aplicación [6].

2.6. ISO 90003

Conjunto de estándares utilizados para el desarrollo, suministro y soporte del software, cuyo propósito es ofrecer una guía de aplicación de la norma 9001 que pretende ser utilizada para demostrar o soportar que la entidad (proyecto/empresa) está en capacidad de desarrollar software con criterios de calidad [2]. La ISO 9001 cuando es aplicada a auditorías del software, tiene cinco categorías [6].

La sección 7.3 de la ISO 90003:2004 cubre una serie de subdivisiones que proporcionan pautas detalladas para las fases del diseño y desarrollo de software, las cuales son:

- 7.3.1. *Design and development planning (Diseño y planeación del desarrollo).*

- 7.3.2. *Design and development inputs (Diseño y entradas del desarrollo).*
- 7.3.3. *Design and development outputs (Diseño y salidas del desarrollo).*
- 7.3.4. *Design and development review (Diseño y revisión del desarrollo).*
- 7.3.5. *Design and development verification (Diseño y verificación del desarrollo).*
- 7.3.6. *Design and development validation (Diseño y validación del desarrollo).*
- 7.3.7. *Control of design and development changes (Cambios del desarrollo y control del diseño).*

2.7. ISO 20000

El objetivo principal de esta norma es el de avalar que la prestación de servicios gestionados de TI (Tecnologías de la información) de una empresa cuentan con la calidad necesaria para brindar dichos servicios a los clientes. Se subdivide en dos partes: "Especificaciones", publicada como ISO 200001:2005, y "Código de buenas prácticas" publicada como ISO 20000-2:2005 [2].

La ISO 20000 es un instrumento potente para los que buscan la dirección del servicio de calidad, si son empresas del sector de la TI, departamentos de la TI o abastecedores internos dentro de alguna otra clase de la organización o hasta otros tipos de organizaciones orientado a servicios. Es relevante notar que, aunque el origen del estándar se arraigue en una intención de manejar servicios de la TI, la versión actual no contiene ningún impedimento para la utilización de ello para otros tipos de servicios.

La ISO 20000 se forma de una serie de documentos. El primer documento, formalmente conocido como ISO/IEC 20000-1 (ISO/IEC 2011), es la única parte que establece requisitos para un sistema de gestión del servicio, por lo tanto, estos requisitos son los únicos que deben cumplirse para obtener la certificación. Los otros documentos, aunque sean muy valiosos y provechosos para poner en práctica el estándar, se pueden considerar complementarios [7].

2.8. PSP, TSP Y CMMI

En esta sección se describe el uso de tres herramientas, que en conjunto permiten a una empresa dedicada al Desarrollo de Software (DS), alcanzar niveles de madurez, se consideran las más actuales y utilizadas en el área de la IS (Ingeniería de Software), la Fig. 2, muestra hablando en términos de calidad y madurez de software, una pirámide de la IS relacionada con los modelos de madurez PSP, TSP Y CMMI, en la que se detalla el nivel de utilidad y de pautas para el uso de las estrategias para la gestión de DS, partiendo desde la vista personal hasta la empresarial; primero en la parte más baja, se debe de comenzar aplicando la gestión a nivel personal, posteriormente a nivel de equipos de trabajo, para poder lograr una integración a nivel organizacional y obteniendo un nivel de madurez como resultado de esta integración de estrategias [8].

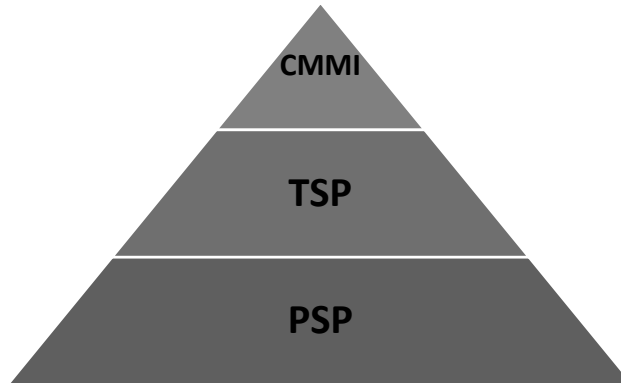


Fig. 2 Pirámide que expresa el nivel de aplicación de las estrategias para la calidad de software PSP, TSP, CMMI [Elaboración propia].

Personal Software Process (PSP)

La demanda de la industria del software para lograr previsibilidad y consistencia frente a los rápidos cambios es significativa. El marco de trabajo PSP ayuda al individuo a satisfacer estas demandas [9]. PSP es un proceso personal que, al estar basado en los principios de mejora, ayuda a la gente a establecer sus metas personales, identificar qué métodos utilizarán, medir su trabajo y analizar los resultados, para ajustar los métodos que utilizan para cumplir sus metas [9].

PSP es un proceso definido para ayudar a realizar mejor el trabajo, cuyo objetivo es obtener y reportar datos precisos y completos del trabajo que se realiza a nivel individual, con el fin de mejorar el proceso individual, afectando de esta manera al desempeño de todo el equipo [9].

PSP amplía el proceso de mejora a la gente que realiza el trabajo de desarrollo de software, concentrándose en las prácticas de trabajo de los ingenieros en una forma individual, enseñando cómo manejar la calidad desde el principio de un producto. PSP métricas propias de cada individuo, que permiten estructurar y ordenar el trabajo del día a día. El resultado del trabajo personal, además puede ser llevado a un trabajo en equipo TSP (Team Process Software), el cual es “comandado” por un sistema de gestión de la configuración y por supuesto, un Jefe de Proyecto quien evalúa los resultados y avances de los miembros del equipo [10].

Algunos de los objetivos del proceso PSP son:

- Introduce la disciplina y la medición del proceso.
- Introduce estimación y planeación.
- Introduce al manejo de calidad.

Los principios de PSP, recopilados por [8] son:

1. Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
2. Usar procesos bien definidos y cuantificados.

3. Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
4. Cuanto antes se detecten y corrijan los errores menos esfuerzo será necesario.
5. Es más efectivo evitar los defectos que detectarlos y corregirlos.
6. Trabajar bien es siempre la forma más rápida y económica de trabajar.

Algunas de las principales características se describen, de manera resumida, en el siguiente listado:

- Concentra la práctica de una forma individual.
- Es de uso personal.
- Se aplica en proyectos pequeños.
- Produce software de calidad.
- Administra tiempo.
- Demuestra cómo manejar el control de calidad.

Los procesos, prácticas y herramientas PSP se diseñaron para ayudar a los ingenieros del software a hacer bien su trabajo, por tanto, permite aplicar métodos avanzados de ingeniería a tareas personales diarias. Proporcionando métodos detallados de planificación y estimación, muestra a los ingenieros cómo controlar su rendimiento frente a estos planes y explica cómo los procesos definidos guían su trabajo [6].

Team Software Process (TSP)

TSP inició como una herramienta capaz de ayudarle a los equipos de gerentes de proyectos, así como a los ingenieros, a organizar y producir proyectos de software a gran escala; se dio a conocer en 1996 y fue desarrollado por el ingeniero y físico Watts S. Humphrey, el cual publicó el primer reporte técnico en el año 2000 [11].

Humphrey buscaba proporcionar a sus estudiantes de ingeniería de software, una visión total del ciclo de vida del software [11].

Esta novedosa herramienta es considerada como una metodología para administrar el trabajo de mejora y desarrollo de los procesos de software, además de garantizar un entorno de trabajo agradable y natural para los equipos. TSP brinda un conjunto de pasos bien estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo, además brinda una ayuda acerca de cómo conformar equipos para el desarrollo de software de calidad [11]. Como toda herramienta, los objetivos que rigen la implementación de TSP [12], se listan a continuación:

- Maximizar la calidad del software en detrimento de los costos.
- Formar equipos que sean capaces de planear y registrar su trabajo, establecer metas bien definidas y sean aptos para realimentar su propio trabajo mediante la medición de éste.
- Brindar un punto de vista a los gerentes y líderes de proyecto acerca de cómo monitorear y como motivar a sus equipos de trabajo para sacar su máximo potencial.
- Establecer una guía para el mejoramiento en organizaciones maduras; así como acelerar la mejora continua de procesos.

Como toda metodología que busca la continua mejora de procesos, TSP posee fases en donde se describen una serie de pautas para ayudar a realizar un buen desarrollo de software por parte del equipo de trabajo. Las fases descritas por [12] son mostradas en la Fig. 3.



Fig. 3 Fases de TSP, descritas en [12], [elaboración propia].

Las principales características de la herramienta TSP, descritas por Humphrey en [11], son:

- Miembros expertos en papeles de liderazgo y pertenencia.
- Relaciones tranquilas y establecidas entre los miembros.
- Los miembros se sienten atraídos por el grupo y son fieles.
- Los valores y metas del grupo son los de sus integrantes.
- Los miembros están motivados por hacer lo que puedan por el grupo.
- La interacción y toma de decisiones tiene lugar en el ambiente adecuado.
- El grupo desea ayudar a cada miembro a adquirir su pleno potencial.

Capability Maturity Model Integration (CMMI)

CMMI es un modelo de procesos que contiene las mejores prácticas de la industria para el desarrollo, mantenimiento, adquisición y operación de productos y servicios. Su enfoque es la mejora de procesos, que provee a las organizaciones de los elementos esenciales para un proceso efectivo y provee guía para la calidad de los procesos. CMMI fue desarrollado por el SEI: Software Engineering Institute, Instituto de Ingeniería de Software [2].

El objetivo de CMMI es crear productos de calidad que permita obtener resultados medibles y repetibles. CMMI mide la madurez del proceso de desarrollo del software en una escala del 1 al 5 (véase Fig. 4) e integra disciplinas como sistemas y software en un solo marco de trabajo, describe formas efectivas y probadas de hacer las cosas, no es un enfoque radical [8].

El modelo CMMI consta de 22 áreas de proceso. Las áreas de proceso de Nivel 2 y 3 son las siguientes: Planificación de Proyecto (PP), Seguimiento y Control de Proyectos (PMC), Desarrollo de Equipos Integrado (IT), Gestión de Riesgos (RSKM), Desarrollo de Requisitos (RD), Gestión de Requisitos (RM), Solución Técnica (TS), Integración del Producto (SI), Verificación (VER), Validación (VAL), Medición y

Análisis (MA), Gestión de la Configuración (CM), Aseguramiento de Calidad de Proceso y Producto (PPQA). CMMI es utilizado en más de 5000 empresas de las cuales 24 son empresas ubicadas en el Perú. En su gran mayoría las empresas peruanas se encuentran en nivel 2 y 3 [8].



Fig. 2 Ilustración de los Niveles de Madurez CMMI. [Recuperado de <http://rupandcmmi.blogspot.com/p/niveles-de-madurez.html>].

En la Tabla 1 se muestra el cuadro comparativo correspondiente a los modelos de calidad a nivel proceso, descritos en este trabajo, por consiguiente, es posible determinar que los modelos relacionados muestran, como principal objetivo, proveer a una empresa desarrolladora de software las armas para construir software de calidad, además de que cuentan con niveles para alcanzar la calidad de una manera progresiva, del mismo modo se pueden adaptar a equipos de trabajo, tanto pequeños como grandes. Lo ideal es conocer más a fondo cada herramienta y realizar un estudio detallado de cada una, es decir, aplicar los modelos presentados en este trabajo o bien, entrevistar a las empresas que los utilizan, para conocer los factores de éxito o fracaso de estos modelos en la industria del software.

Por otro lado, un ejemplo de modelos para alcanzar un nivel de calidad de tipo híbrido, es decir modelos de calidad a nivel proceso, producto y/o funcionalidad, Luis E Mendoza, ver [13], propone MOdelo Sistémico de CALidad (MOSCA), para evaluar la calidad de los sistemas de software, integrando el modelo de Calidad del Producto y el modelo de Calidad del Proceso, MOSCA fue probado en dos empresas venezolanas desarrolladoras de software, utilizando el Método “Análisis de Características por Caso de Estudio”, con 3 fases para ir incrustando la calidad en los sistemas de software. Lo anterior para destacar otra oportunidad de estudio, es decir, integrar aquellos modelos que permitan implementar la calidad de software en sus tres horizontes; nivel de proceso, producto o calidad en uso.

3. Conclusiones y trabajos futuros

Es evidente que los procesos de desarrollo en una industria emergente, como el software, que continuamente aporta nuevos escenarios tecnológicos, deben de contar con una herramienta de evaluación de la madurez de sus procesos, para que continuamente este siendo evaluada; de tal forma que la entidad pueda nutrirse y así crecer y aportar conocimiento al resto del panorama industrial.

Tabla 1. Cuadro Comparativo: Modelos de Calidad Nivel Proceso.

	ISO 9003	ISO/IEC 2000	CMMI	TSP	PSP
Objetivo	Conjunto de estándares utilizados para el desarrollo, suministro y soporte del software.	Avalar que la prestación de servicios gestionados de TI de una empresa cuenta con la calidad necesaria para brindar dichos servicios a los clientes.	Modelo de procesos que contiene las mejores prácticas de la industria para el desarrollo.	Metodología para administrar el trabajo de mejora y desarrollo de los procesos de software.	Proceso personal que ayuda a la gente a establecer sus metas personales.
Ámbito	Organizacional	Organizacional	Organizacional	Equipo	Personal
Niveles/ actividades para alcanzar madurez y calidad	cinco categorías	Documentación	Escala del 1 al 5	Procesos, prácticas	Procesos, prácticas
Tipo de proyectos aplicables	Grandes	Grandes	Grandes	Grandes	Pequeños
Certificación	SI	SI	SI	SI	SI

	ISO/IEC 15504	BOOTSTRAP	DROMEY	IEEA/E/A 12207	COBIT
Objetivo	Proponer y desarrollar un estándar de evaluación de procesos de software.	Metodología de valoración de la calidad que permite la mejora de procesos a partir de 6 actividades básica.	Es un modelo adaptable a evaluar varias etapas del proceso de desarrollo.	Marco de trabajo para el ciclo de vida del desarrollo de software a partir del planteamiento de procesos, actividades y tareas.	Marco influyente para el control y el gobierno de las TI.
Ámbito	Pequeñas y medianas empresas	Equipo	Organizacional	Organizacional	Equipo
Niveles/ actividades para alcanzar madurez y calidad	6 niveles	6 actividades	Tres modelos distintos para cinco etapas 5 etapas.	Se compone de 3 partes	Siete criterios
Tipo de proyectos aplicables	Grandes /pequeños	Grandes	Grandes	Grandes	Grandes
Certificación	SI	-	-	-	SI

En este artículo, se proporciona una revisión breve de herramientas que permiten ir evaluando la calidad de los procesos de desarrollo de software. Dentro de la literatura se encuentran algunos trabajos como en [2] and [5], que se ha encargado realizar un revisión más exhaustiva a nivel de proceso, producto y uso.

Como conclusión es posible afirmar que una empresa, del ámbito del desarrollo de sistemas informáticos, que no cuente con algún sistema de gestión de calidad o de evaluación de sus procesos de desarrollo de software tiene la tarea difícil de situarse en un mercado competitivo para tener éxito en cualquier licitación pública o privada; por tanto, se vuelve indispensable que conozca aquellas herramientas que le permitan iniciarse en el camino hacia la calidad en el desarrollo de software.

Un trabajo futuro para esta investigación es aplicar los modelos presentados en este trabajo o bien, entrevistar a las empresas que los utilizan, para conocer los factores de éxito o fracaso de estos modelos en la industria del software.

Referencias

1. López-Echeverry, A.M., Cabrera, C., Valencia-Ayala, L.E.: Introducción a la calidad de software. *Sci. Tech.* 2(39), 326–331 (2008)
2. Callejas-Cuervo, M., Alarcón-Aldana, A.C., Álvarez-Carreño, A.M.: Modelos de calidad del software, un estado del arte. *Entramado* 13(1), 236–250 (2017)
3. ISO 15504 Norma de desarrollo de software SPICE ISO/IEC 15504, <https://www.normas-iso.com/iso-iec-15504-spice/>
4. Estándares de Calidad en el Mercado Estándar, (1993)
5. Das, S.R., Yaylacicegi, U., Canel, C.: Using ISO 90003 for software ‘design and development’ in large virtual teams. *Ind. Manag. Data Syst.* 108(6), 775–793 (2008)
6. Cots, S., Casadesús, M., Marimon, F.: Benefits of ISO 20000 IT service management certification. *Inf. Syst. E-bus. Manag.* 14(1), 1–18 (2016)
7. Chavarría, A.E., Oré, S.B., Pastor, C.: Aseguramiento de la calidad en el proceso de desarrollo de software utilizando CMMI, TSP y PSP. *RISTI - Rev. Iber. Sist. e Technol. Inf.*, vol. 2016, no. 20, pp. 62–77 (2016)
8. Kamatar, J., Hayes, W.: Experience report on the personal software process. *IEEE Softw.* 17(6), 85–89 (2000)
9. Humphrey, W.S.: Introducing the personal software process. *Ann. Softw. Eng.* 1(1), 311–325 (1995)
10. Humphrey, W.S.: Team software process (2000)
11. Díaz Sánchez, D.Y., Maldonado Reyes, I.M., Reveles Pérez, O.D., Escamilla Camargo, A.: TSP-Team Software Process (2017)
12. Mendoza, L.E., Pérez, M.A., Grimán, A.C.: Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software / Prototype of Software Quality Systemic Model (SQSM) vol. 8, no. 3, Centro de Investigación en Computación, IPN (2005)

Evaluation of Five Classifiers for Depression Episodes Detection

Susana L. Pacheco-González¹, Laura A. Zanella-Calzada¹,
Carlos E. Galván-Tejada¹, Nubia M. Chávez-Lamas²,
J. Francisco Rivera-Gómez¹, Jorge I. Galván-Tejada¹

¹ Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Mexico

² Universidad Autónoma de Zacatecas, Unidad Académica de Odontología, Mexico
{lzanellac, ericgalvan, jf_riverag, gatejo, nubiachavez}@uaz.edu.mx

Abstract. Depression is a mental disorder manifested through a set of psychological and physical symptoms, such as the presence of sadness, apathy, hopelessness and irritability, among others. According to the World Health Organization (WHO), depression is affecting more than 300 million people worldwide, presenting a prevalence between 3 and 21%. One of the main problems of this high prevalence is the incorrect classification of patients, since many cases are false positive and false negative diagnoses. In this work it is proposed the study of the behavior of five different classification techniques, random forest (RF), conditional inference trees (cTree), K-nearest neighbor (K-NN), support vector machine (SVM) and Naïve Bayes, to identify depressive states through the motor activity of patients contained in the Depresjon dataset. The activity of this dataset is acquired through the smart watch “Actigraph”, based on actigraphy. The evaluation of these classification techniques is finally performed in terms of sensitivity, specificity, the receiver operating characteristic (ROC) curve and area under the curve (AUC), to know their performance to automatically detect depressive patients. The results shown values of sensitivity, specificity and AUC, statistically significant, specially for the RF method, which presents sensitivity = 0.8148, specificity = 0.8158 and AUC = 0.8314. Therefore, it is concluded that these classifiers are able to distinguish patients with depression from controls, based on their motor activity, allowing the development of a non-invasive diagnosis tool to support specialists in the correct diagnosis of depression.

Keywords: depression, motor activity, classification, random forest, conditional inference trees, K-nearest neighbor, support vector machine, naïve Bayes.

1 Introduction

Depression is a mental disorder which is manifested through a set of psychological and physical symptoms. Among these symptoms are present the predominance

of feelings related to sadness, apathy, hopelessness and irritability, besides the deterioration in personal appearance, crying and insomnia. This disorder can also be accompanied by other emotional and / or physical illnesses, such as anxiety, alcohol and other substance abuse, systematic illnesses, eating disorders and even some personality disorders [5].

According to the World Health Organization (WHO), in 2017 the depression was present in more than 300 million people worldwide, having a current prevalence between 3 and 21 %. Mainly, this condition affects people from 15 to 45 years old and, in most countries, it occurs twice as many women as men. In addition, this condition appears as the second cause of death in people aged 15 to 29, causing about 800,000 suicides per year [11].

This disorder is regularly classified into three main categories, mild, moderate or severe, which help to make an adequate diagnosis. These categories depend on the duration with which the symptoms occur and can range from transient to persistent (the presence lasts for months or years) and, according to this classification, the disease can rebound in different ways, interfering with the work, the school, family, society, among others [8].

The diagnosis of depression also allows to receive a more appropriate treatment according to the degree to which it occurs. However, according to the literature, about 56 % of the cases are not diagnosed due to the lack of resources and trained health personnel, together with the inaccurate clinical evaluation and the mythification that is had of this disease by of patients, which is based on stigma related to mental disorders caused by lack of education.

On the other hand, the problem of false positives and false negatives in the diagnosis of depression continues to present itself in countries of all types of income in a significant way, despite the extensive study that has been focused on this disease, causing patients with depression do not receive any type of treatment while healthy patients or patients with a condition other than depression are diagnosed as positive and are subject to antidepressants [1].

Due to this diagnostic problem, different artificial intelligence (IA) tools have been implemented that can contribute to improve the detection techniques of this disorder. The main purpose of this approach is the development of tools that allow to automatically identify when a patient presents depression from certain type of information provided by the patient. To be able to carry out this task, the AI tools are based on a previous training, which consists in submitting an algorithm to an automatic learning that consists in the search of relationships of a considerable number of examples of cases similar to those that they want to classify, with the real diagnosis of those examples, being information of patients with depression specifically for this case [14].

Based on this, the main contribution of this work is to study and compare the behavior of different types of classification techniques to predict depressive states through the activity of patients, measured with a smart band accelerometer, this evaluation in terms of sensitivity and specificity, as a receiver operating characteristic (ROC) curve and area under the curve (AUC), let to know the performance to maximize the detection of depressive or not depressive states

depending on what researchers want to develop for a particular application. Additionally, this comparison allows the development of a real-time non-invasive diagnosis. Automatic depression states can increase significantly the development of an effective treatment and contribute to the prevention of this type of psychopathology.

This work is organized as follows. In section 2 is presented the materials and methods evaluated in this work, random forest (Rf), conditional inference trees (cTree), K-Nearest Neighbor (K-NN), support vector machine (SVM) and Naive Bayes. In section 3, the results obtained are shown. Finally, in section 4 these results are discussed and the conclusions are briefly described.

2 Materials and Methods

This section presents the detailed explanation of the stages proposed by this work, shown in Figure 1. Initially, a data preprocessing is applied to extract a set of data to represent the behavior of the patients. These data are subsequently subjected to five classifiers, RF, cTree, K-NN, SVM and Naive Bayes in order to find a model that allows to the automatic classification of patients with presence of depression. Finally, the classification obtained by the different methods is validated through a set of statistical metrics, such as the ROC curve, AUC, sensitivity and specificity.

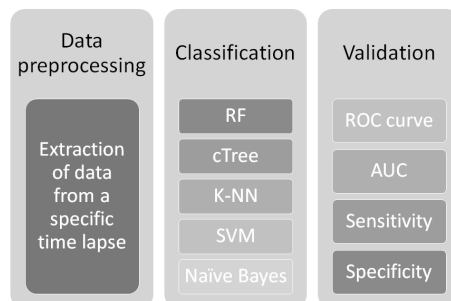


Fig. 1. Flowchart of the methodology proposed.

2.1 Data Description

For the development of this work, the Depresjon dataset is used, which is a collection of data contained by the information of the motor activity from patients with presence and absence of depression. These data have been deposited in the “control” and “condition” repositories of the dataset, being possible to access in <http://datasets.simula.no/depresjon/> and / or can be directly downloaded from <http://doi.org/10.5281/zenodo.1219550>.

The information contained in the dataset is obtained from 55 patients, 32 controls (absence of depression, represented with 0) and 23 cases (presence of depression, represented with 1). For the acquisition of the data, the actigraph watch “Actiwatch” (model AW4, developed by Cambridge Neurotechnology Ltd, England) is used, which consists of patients carrying it on their left wrist for a series of days during the 24 hours. The Actiwatch measures the activity levels with a sampling frequency of 32 Hz, recording movements over 0.05 g. Movements are represented as voltage, which are stored as an activity count in the memory of the Actiwatch, being the number of counts proportional to the intensity of the movement. These activity counts are recorded in on minute lapses [3].

2.2 Data Preprocessing

For the preprocessing of the data, a specific time lapse was selected for each patient from the total data acquired. That is, for each observation, 60 samples are taken from a lapse of one hour, having a sampling frequency of one minute. These samples are selected within the time range of 9 pm to 5 am.

It is important to note that this time period is selected due to its clinical implication, which refers to the fact that subjects with depression tend to present a greater amount of motor activity during that time, while control subjects do not, meaning significant information for the distinction of case subjects.

2.3 Classification Methods

For this work five methods were selected to study the performance of different categories of classifiers to detect depression episodes, represented by an outcome with two possible values, 0 (controls) and 1 (cases). RF and cTree follow a decision based approach which helps to take into account the time dependence, on the other hand, K-NN and SVM is an instance based method to include all the observation as a unit, while Naive Bayes is a probabilistic approach.

Random Forest This method is a well-know classifier proposed by Breinman et al. [2], which uses two levels of randomness for the construction of the trees, beginning with a bootstrapped version of the training data, known as bagging, where a subset of the training data is used for each tree, based on the principle of replacement and the remaining data is used to estimate the error, through the out-of-bag (OOB) error. In the second level, a subset of features is randomly selected and added to each node throughout the growth of the decision trees. For each node, the best feature among a random subset of features is selected in order to reduce the label error. This technique bases its classification technique on taking the majority vote from all the decision trees. This process is recursively repeated until the trees reach a defined depth or the number of samples in a node does not exceed a threshold [15,10].

Conditional Inference Trees This method based on recursive partitioning, which is a fundamental tool to explore the structure of the data and developing decision rules for predicting a categorical or continuous outcome. The conditional distribution of statistics to measure the relationship between the input features and the output is the bases for the unbiased selection among covariates measured in different scales. In an algorithm that performs recursively partitions, a sample is developed using non-negative integer valued case weights. In cTree, two main steps are involved, (1) feature selection and (2) splitting. The step 1 consists in the selection of the covariate of strongest association with the output for splitting. Then, in step 2 a permutation test framework is developed to find the optimal binary split for the selected covariate in step 1. These steps are repeated recursively until the global null hypothesis of the independence between the output and any of the features cannot be rejected at a pre-specified threshold, which can be the p-value, to mention an example [12].

K-Nearest Neighbor This method present a non-parametric approach that has been widely used in different statistical applications since the early 1970's. The basis of this algorithm consists on, from the calibration dataset, a group of K samples is found, which are nearest to unknown samples. To achieve this goal, the Euclidean distance, $\|\vec{x} - \vec{y}\|$, between a given set of queries and the inputs, is calculated, and the K closest input points are identified for each query [6]. From the these K samples, the output of the unknown samples are determined through the calculation of the average of the input features. The parameter K is determined based on several calculations. Therefore, as a result of this classifier, the k is a key element in the performance of the K-NN [9].

Support Vector Machine This method is based on statistical learning theory proposed by Vapnik et al. [13], where the goal is to achieve structural risk minimization. A learning strategy to keep the empirical risk value fixed and minimize the confidence range is applied in this approach. Its basis is obtained from the optimal classification hyperplane with linear separability, which is mainly aimed in binary classification problems. The objective is to find a hyperplane to correctly separate the two types of outputs without errors, while the separated data points farthest from the classification surface are kept. To solve this problem and get the classifier, a constrained quadratic programming problem is constructed based on $\min \frac{1}{2} \|w\|^2$, $\text{sty}_i(w \cdot x_i + b) \geq 1$, $i = 1, 2, \dots, n$. According to the statistical learning theory, the error-free separation helps to ensure that the empirical risk is minimized, the distance between classifications is maximized and the confidence range of the generalization bound is minimized, minimizing the real risk. Therefore, SVM provides a good generalization ability.

Naïve Bayes This classifier is based on the Bayes theorem, which consists on Equation 1, where $G = G_i$ are the possible classes for a data point X and $Pr(G = G_i|X)$ represents the posterior probability based on prior knowledge combined with the observed data:

$$Pr(G = G_i|X) = \frac{Pr(X|G = G_i)Pr(G = G_i)}{Pr(X)}. \quad (1)$$

The Naïve Bayes model assumes that the inputs are conditionally independent in each class, and the classifier does not consider correlation between features, lowering the variance while the bias is increased and the overfitting is avoided on the training set. The classification of a vector X with d features is provided by taking the maximum posterior probability over all classes and applying the Bayes' theorem, as shown in Equation 2:

$$\underset{G_i \in G}{argmax} Pr(G = G_i) \prod_{j=1}^d Pr(X = X_j|G = G_i). \quad (2)$$

Then, all the information used to classify an arbitrary data point can be represented in a set of vectors, which is useful to carry out the classification.

2.4 Validation

For the validation stage, three parameters were measured, the AUC as a quantity value of the ROC curve, the specificity and the sensitivity.

The ROC curve is a metric widely used for the evaluation of binary classification problems, allowing to the graphic interpretation of the results based on the shape of the curve, being possible to provide its quantitative value with the AUC [7]. This parameter allows to measure the classification precision of the model through the sensitivity and the specificity.

The sensitivity is defined as the proportion of samples with a positive condition that are correctly classified. This metric can be calculated with Equation 3, where TP represents the true positives and FP represents the false positives [4]:

$$sensitivity = \frac{TP}{TP + FP}. \quad (3)$$

The specificity is defined as the proportion of samples with a negative condition that are correctly classified. It can be calculated with Equation 4, where TN represents the true negatives and FN represents the false negatives [4]:

$$specificity = \frac{TN}{TN + FN}. \quad (4)$$

Finally, the sensitivity and the specificity in conjunction represent the decision threshold of the ROC curve. The AUC can be calculated through trapezoidal integration, as shown in Equation 5:

$$AUC = \sum_i (1 - \beta_i \cdot \Delta\alpha) + \frac{1}{2}[\Delta(1 - \beta) \cdot \Delta\alpha], \quad (5)$$

where $\Delta(1 - \beta) = (1 - \beta_i) - (1 + \beta_{i-1})$ and $\Delta\alpha = \alpha_i + \alpha_{i-1}$.

3 Results

The results obtained from the performance of the classifiers are presented. Figure 2 shows the ROC curves calculated based on the performance of each classifier. The ROC curves (a) and (b), which correspond to RF and cTree, respectively, present very similar curves, besides being the ones that present the best behavior in conjunction with the curve (d), which corresponds to SVM. The curve (c), which corresponds to K-NN also presents an adequate behavior, although a little less than the previous curves; while the curve (e), which corresponds to Naïve Bayes, is reduced significantly compared to the others.

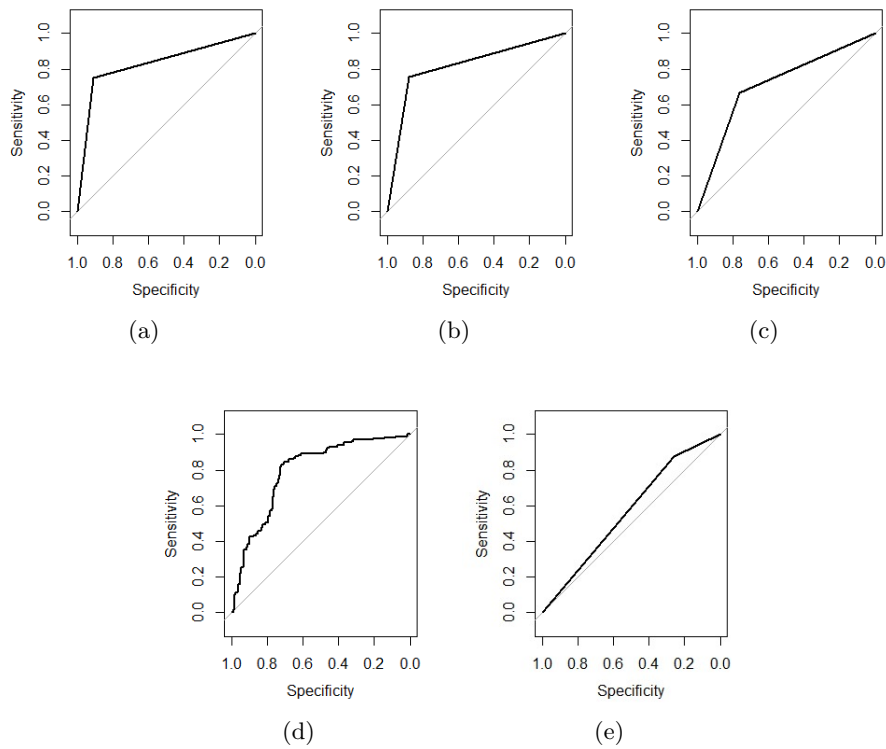


Fig. 2. ROC curves obtained using the (a) RF, (b) cTree (c) K-NN, (d) SVM and (e) Naive Bayes.

The quantitative results that justify the behavior of the ROC curves are presented in Table 1, where are shown the values obtained from the statistical metrics measured for each technique, related to the identification of patients with depression. As is possible to observe, RF presents the highest values for each metric, obtaining a sensitivity of 0.8148, specificity of 0.8158 and AUC

of 0.8314. Naïve Bayes presents the lowest AUC with a value of 0.5677 and specificity with a value of 0.4606; however, its sensitivity presents a value of 0.7419, which is higher than the SVM sensitivity, since it obtained a value of 0.6364, being the lowest sensitivity.

Classifier	Sensitivity	Specificity	AUC
RF	0.8148	0.8158	0.8314
cTree	0.7383	0.8090	0.8183
K-NN	0.7614	0.6693	0.7153
SVM	0.6364	0.7200	0.7924
Naïve Bayes	0.7419	0.4606	0.5677

Table 1. Values obtained from the validation.

4 Discussion and Conclusions

In this work is presented the evaluation of five well-known classifiers. These classifiers are different approaches, probabilistic, decision based, deterministic, etc. allowing the comparison in terms of sensitivity and specificity, to overcome the depression episodes automatic detection using an activity signal. this evaluation allows to conclude several specific points described below:

- The ROC curve allows to conclude that Naive Bayes is the one with worst AUC, however, one possible way to improve the behavior of this classifier is to increase the number of observations.
- Decision based approaches acquire the best AUC, however is necessary a deep study of overfitting, given that is a common issue with this type of algorithms.
- Random Forest outperforms other approaches, including other type of decision based algorithms, nevertheless is one with more computational cost when number of trees are increased.
- Random Forest outperforms in terms of sensitivity and specificity other classifiers, however, in medical diagnostics commonly is more important to increase sensitivity to find true positive states. Therefore, K-NN is one possible solution at a lower computational cost, being the second one in terms of sensitivity.

Then, it is important to remark that despite the classifiers it is possible to develop an automatic detection of depressive episodes using the motor activity signal on one hour intervals records without other type of processing. The results obtained probe that independently, there is a particular behavior on patients with depression in the body movements that can be modeled by these type of classifiers.

Therefore, this work allows the development of efficient tools for automatic diagnosis of depressive episodes to help specialists in mental illness to recognize the health status of patients based on the motor activity recorded by commonly used smart bands.

References

1. Alcalá-Rmz, V., Zanella-Calzada, L.A., Galván-Tejada, C.E., García-Hernández, A., Cruz, M., Valladares-Salgado, A., Galván-Tejada, J.I., Gamboa-Rosales, H.: Identification of diabetic patients through clinical and para-clinical features in Mexico: An approach using deep neural networks. *International journal of environmental research and public health* 16(3), 381 (2019)
2. Breiman, L.: Random forests. *Machine learning* 45(1), 5–32 (2001)
3. Garcia-Ceja, E., Riegler, M., Jakobsen, P., Resen, J.T., Nordgreen, T., Oedegaard, K.J., Fasmer, O.B.: Depresjon: A motor activity database of depression episodes in unipolar and bipolar patients. In: *Proceedings of the 9th ACM on Multimedia Systems Conference. MMSys'18*, ACM, New York, NY, USA (2018), <http://doi.acm.org/10.1145/3204949.3208125>
4. Hanley, J.A., J, M.B.: The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology* (1982)
5. Instituto Nacional De Psiquiatría Ramón De La Fuente Muñiz: ADEP Ayuda para Depresión (2019), <https://ayudaparadepresion.org.mx/>
6. Ryoo, J., Arunachalam, M., Khanna, R., Kandemir, M.T.: Efficient k nearest neighbor algorithm implementations for throughput-oriented architectures. In: *2018 19th International Symposium on Quality Electronic Design (ISQED)*. pp. 144–150. IEEE (2018)
7. Saito, T., Rehmsmeier, M.: Precrec: fast and accurate precision–recall and roc curve calculations in R. *Bioinformatics* 33(1), 145–147 (2017)
8. Ministerio de sanidad, s.s.e.i.: Guía de Práctica Clínica sobre el Manejo de la Depresión en el Adulto. Tech. rep., Galicia (2014), http://www.guiasalud.es/GPC/GPC.534_Depresion_Adulto_Avaliat_compl.pdf
9. Thanh Noi, P., Kappas, M.: Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors* 18(1), 18 (2018)
10. Vekeman, F., Piña-Garza, J.E., Cheng, W.Y., Tuttle, E., Giguère-Duval, P., Oganisian, A., Damron, J., Duh, M.S., Shen, V., Saurer, T.B., et al.: Development of a classifier to identify patients with probable Lennox-Gastaut syndrome in health insurance claims databases via random forest methodology. *Current medical research and opinion* (just-accepted), 1–1 (2019)
11. WHO: WHO — Depression (2017), https://www.who.int/mental_health/
12. Yang, J.J., Lakte, P.T.A., Kim, S.H.: Use of conditional inference trees for evaluating the effect of reclaimed asphalt pavement content and binder grade on the dynamic modulus of asphalt concrete mixtures. *International Journal of Pavement Research and Technology* (2018)
13. Ye, K.: Key feature recognition algorithm of network intrusion signal based on neural network and support vector machine. *Symmetry* 11(3), 380 (2019)
14. Zanella-Calzada, L.A., Galván-Tejada, C.E., Chávez-Lamas, N.M., Gracia-Cortés, M., Magallanes-Quintanar, R., Celaya-Padilla, J.M., Galván-Tejada, J.I., Gamboa-Rosales, H.: Feature extraction in motor activity signal: Towards a depression episodes detection in unipolar and bipolar patients. *Diagnostics* 9(1), 8 (2019)

Susana L. Pacheco-González, Laura A. Zanella-Calzada, Carlos E. Galván-Tejada, et al.

15. Zhu, X.e.a.: Random forest based classification of alcohol dependence patients and healthy controls using resting state mri. *Neuroscience letters* 676, 27–33 (2018)

Generation of Non-Coordinate Navigation Knowledge from a Flow of Input Views

Josué Figueroa-González, Georgii Khachaturov, Juan M. Martínez-Hernández

Metropolitan Autonomous University - Azcapotzalco, Mexico City, Mexico
{jfgo, xgeorge}@azc.uam.mx, martinez.juan.hdz@gmail.com

Abstract. An approach to automatic generation of a network of reference ‘views’ specifically structured for non-coordinate robot navigation is presented. The reference views are selected from the input flow while the robot moves along a random continuous trajectory in its environment. The network is organized as an atlas that represents inner model of the robot environment. It accumulates a ‘rich’ navigation knowledge extracted from the flow of ‘poor’ views. No coordinate transformation is involved, but fundamental topological concepts: the continuity of a robot trajectory and discontinuities (‘heavy changes’) detected in the input flow. Avoiding any explicit coordinate transformation becomes possible using a local inversion of the knowledge hidden in the relation ‘robot control – change of view’. Both techniques – the inversion and detection of heavy changes – are specific for a particular robot architecture. The approach was verified on a virtual static 2D-workspace.

Keywords: cognitive robotics, learning workspace from views, non-coordinate environment modeling, visual navigation.

1 Introduction

This work is a step in a wider research about non-coordinate robot navigation. It presents a biologically inspired approach that does not use coordinates of any kind, neither for pointing a navigation goal, nor for an inner representation of navigational knowledge. ‘Non-coordinate’ navigation used in this work, intends to emphasize the fact that the commonly used term ‘non-Cartesian’ can still refer to a coordinate system. The architecture proposed here covers both, the issue of inner representation of geography, as well as learning the geography process during exploration of the environment by robot.

Fig. 1 clarifies our goal. It shows a workspace with a virtual robot. The workspace is static, bi-dimensional, and populated with obstacles. Fig. 1a shows a supervisor’s view of an instance of such workspace with the robot shown as a dark triangle. If the robot had a colour camera, its current view would be as shown in Fig. 1b. But instead, robot has a sensor of 1D range images that scans all directions in a 1D field of view and measures the distance to the first obstacle along each direction, so the robot view for the same scene is shown in Fig. 1c.

Note that the distances represented by such views are not crude, but somehow transformed due to technological reasons related to OpenGL.

The single view in Fig. 1a provides a human with full information about geography of the environment, but the robot, at any time, has a much poorer input: the one similar to the shown in Fig. 1c. By processing a flow of such ‘poor’ views, is constructed a data structure that contains a complete information about the geography, equivalent to the one in Fig. 1a.

The inner workspace model of the robot is constructed as a set of reference views from the input flow and is organized as an atlas of ‘neighbourhoods’.

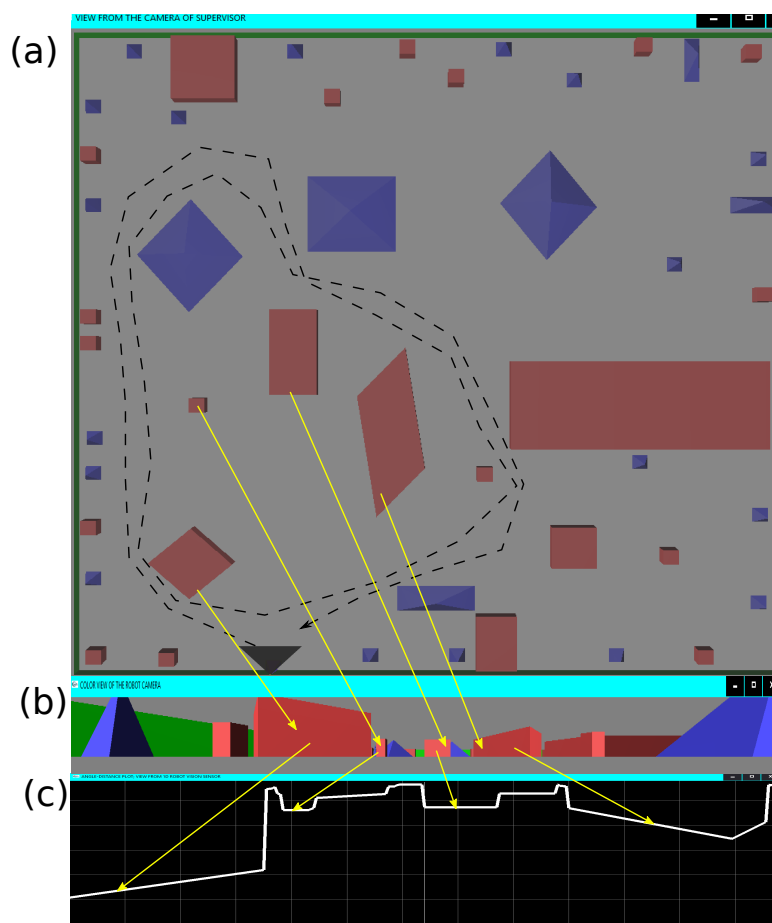


Fig. 1. Three views for the same “robot-in-environment” scene: (a) A supervisor’s view; the robot position and orientation is shown by a dark triangle at the bottom; the dashed line is commented in Section 4. (b) A robocentric view by a colour camera. (c) A robocentric view from a sensor of 1D range image. The lines between (a) and (b), and between (b) and (c) connect the same objects on corresponding views.

The process of learning the geography is based on driving the robot along a continuous random trajectory and an analysis of the input flow of such views; at the beginning, the inner model of workspace is initialized to a single reference view that corresponds to an initial robot position on the scene. A test over any pair of views, being applied to the pairs of adjacent views of the input flow, is aimed to show semantic discontinuities in the flow. The positive responses of the test are called ‘heavy changes’. Each neighbourhood is formed by a set of views with no heavy change between them; the data structure that represents a neighbourhood contains also information about entrances from and exits to other neighbourhoods. This network is a dynamic structure: it is subjected to the inclusion of new reference views, linking them to neighbourhoods, adding new neighbourhoods into atlas, adding links (entrances or exits) between adjacent neighbourhoods and merging neighbourhoods after detecting a loop closure of the robot trajectory in its workspace.

The network represents a graph-based component of the robot navigation knowledge. It can be used for solving non-local navigation tasks, meanwhile for the local navigation a vision-guided control is assumed. Retrieving a command for a desirable change of views is performed by the Kohonen associative memory, [15]. Training the memory is accomplished by a set of ‘cause-effect’ pairs {robot command; change of views}.

Navigation assumes self-localization, in contrast to SLAM, localization in the scope of presented approach is non-coordinate; it is performed as a permanent task that at any moment generates a pair {Current Neighbourhood (CN); Current Reference View (CRV)}, where the components, respectively, stand for IDs of the neighborhood which the current robot view belongs, and a reference view in CN from which the current view can be reached by a robot control command. An estimate of the command is constructed by associative memory.

For the sensor that produces images like the ones in Fig. 1c, estimation of the robot control command for a desirable change of views is as follows: The associative memory is represented as a vector with components indexed by possible distances to objects in a view. The vector representation is required because each object in the field of view responds non-linearly to a robot control according to the distance to this object. For two views that determine a required change of robot state, any pair of matching objects in them contributes with its own impact into the estimate of the command. This impact is computed by a component of the vector of associative memory that corresponds to a distance to this particular object. Amount of all impacts gives the required estimate.

Heavy changes are the milestones for the non-coordinate navigation. Detection of heavy changes is sensor-dependent. A corresponding technique for the views like in Fig. 1c was presented in [14]. The relation of heavy change helps to associate the robot views with some neighborhoods of the atlas: two close views generated on a continuous robot trajectory belong to the same neighborhood if there is no heavy change between them, and vice versa.

This work focuses on the logic of creating the network of reference views, this approach was tested using a new version of the software presented in [14].

The rest of the paper is organized as follows: Section 2 offers a review of previous works. Section 3 describes a logic for learning the inner model of workspace. Section 4 describes the structure of software and experiments on it. Section 5 contains conclusions.

2 Related Works

2.1 Non-Coordinate Navigation

The attempts to understand how the human brain tackles the navigation tasks have been undertaken in numerous works during several decades, mainly in psychology. In robotics, it is an active theme for more than two decades.

An approach to navigation of autonomous systems in semi-structured environments using natural language is based on a model called Generalized Grounding Graphs (G3) [19]. The robot must be trained beforehand for recognizing basic elements like: events, objects, places and paths. Then they can be used to express a navigational goal in natural language as “go to next corner of the room”. Closer ideas can be found in [2], where the robot must complete a task which involves identifying an object and a position in an unknown environment. Once the task and object are identified, the robot must go to a specific location. In this work the robot learns using imitation learning. In both works, navigation does not deal with a coordinate space.

A graph based model for a non-coordinate vision-guided navigation was proposed in [13]. Other works on this topic are [8,9,18]. The model presented in [13] (so called GT-model) is formed by two graphs: G, which stands for the robot vision-to-action knowledge, and T that represents a geography or administrative system imposed on G. G alone is quite similar to the view graph of [8,18], but being enriched by geography graph T it obtains some new properties.

In formal terms, a GT-model is defined as pair $\{G, T\}$, where the components satisfy the following properties:

- $G = (V_G, E_G)$ is a directed graph with non-negative weights assigned to its edges; intuitively, any node of V_G stands for a set of close robot views and implicitly it represents a subset in a space of all robot states; the edges of E_G stand for some control rules that drive robot from one set of close views to other.
- $T = (V_T, E_T)$ is a directed graph with an unique source-node whereas the set of its terminal nodes coincides with V_G ; the nodes of V_T stand for geographic entities of the robot space, whereas the edges of E_T are just inclusions indicating that one entity belongs to some other.

In these terms, for example, “Move robot from the Mexico City downtown to the Computer Research Center”, means to drive robot from one set of views (downtown) to another (CRC) by means of the rules from E_G . Then, a formalization of a non-coordinate navigation problem is as follows. Let $\alpha, \beta \in V_T$ be geographic entities and $V_\alpha, V_\beta \subset V_G$ be their respective domains. The problem

is: Find a path in G with minimal summary weight that begins inside V_α and ends inside V_β . This problem is called the extended shortest path problem (*espp*). So an *espp* means search of a best route that connects two sets from V_G , however not arbitrary ones but only those represented by some nodes of the geographic graph T .

It turns out, [13], that a navigational problem in the form of *espp* under some natural assumptions can be solved by a dynamic programming algorithm of lower computational complexity than for a classic shortest path problem for two nodes of graph G .

In spite of some advantages of the approach based on GT-model, its application to robotics for the years after publication of [13] remained as an abstract theory than a practical instrument due to the difficulties related to learning the lowest, non-verbalizable level of geography.

Indeed, consider a geographic item that has an explicit name like North America, Mexico City, The National Polytechnic Institute of Mexico, etc. It is technically clear how to introduce such an item into a GT-model. But for the lowest geographic level, each item should be represented by a data structure that does not have neither a distinctive shape, nor a name. In this respect, this work together with [14] represents a novel step to overcome obstacles for future practical implementation of the non-coordinate approach to navigation of [13].

Note that in contrast to the verbalizable levels, the lowest-level geography model is sensor-dependent: the higher levels can be the same, say, for a robot, a blind person, as well as for a human who does not suffer any illness of sight; the lowest level geography depends: for a blind person – on the stick which he/she uses for exploration of the environment, for a normal person – on his/her vision. This is why learning geography of the lowest level cannot be done in the same way for all kinds of robots. But some common principles can be found.

2.2 SLAM and the Navigation Based on a Sketch of Path

The research line called Simultaneous Localization and Map Building (SLAM) has a story of more than a three decades [4–6, 11]. The aim of this approach is to convert a set of robocentric views into a single map. This ‘map’ is understood in a traditional, Cartesian meaning so that a human could read and use it. The SLAM techniques involve statistical methods including extended Kalman filter [3] and Rao-Blackwellized particle filters [10]. They allow feeding the map creation while the robot moves smoothly.

An approach to robot guidance in an indoor environment is presented in [1]. As a human-robot interface, it makes use of a sketch of a required robot path; the robot task in a sense is an inversion of SLAM: starting from a poor quality map (sketch), identify its landmarks in the robocentric views, and use them for guidance. The sketch can be regarded as a single-task-oriented, “light” navigational model of environment. The recognition is simplified due to the indoor specific.

We adopt here the, employed in many SLAM works, idea about treating loops, [7, 12, 17]. Detection of a loop-closure in a robot trajectory increases

drastically the precision of the map under construction. In the scope of our approach, detected loops allow to improve adequateness of an inner model of the workspace.

Note that a single 2D view in Fig. 1a is equivalent to a 2D map, however this way of presentation of the workspace is chosen here not because of some limitations of our approach but just for illustrative and testing purposes. A map like in SLAM not always allows an adequate representation of geography of a robot environment. For example, it is inadequate for an environment similar to a 3D-maze, or for a mountain cave of a complex structure. On the opposite, the approach presented in this work is not limited to a 2D-environment.

Nevertheless, we believe that ideas of SLAM are very fruitful to be combined in future with our approach especially for facing the robot dynamics that are disregarded yet in our experiments.

2.3 Detection of Heavy Changes

Robot state for the workspace of Fig. 1 can be regarded as the pair {robot position; robot orientation}. From a topological point of view, while robot moves smoothly in workspace, a path in the space of robot states is mapped continuously into a space of robot views. Some qualitative changes in the flow of views are declared as Heavy Changes (HCs). Detection of HCs helps to organize the inner model of workspace. As it was mentioned in section 1, a particular technique for detection of HCs depends on the robot architecture. What a HC means depends also on a semantic interpretation of objects in a current view: for instance, in the flow of range images like Fig. 1c, the dynamics of distant objects is not as relevant for HC as that of close objects. Then, detection of HCs in a border zone between different neighbourhoods provokes a natural instability of outcome. In these circumstances, providing the robustness of the atlas of neighbourhoods formed by reference views is not an elementary issue.

A method for tackling this issue was presented in [14] for the virtual world of Fig. 1. There the flow of 1D range images is subjected to the processing that combines a transformation of each image into a string of symbols of an alphabet with comparison of such strings by means of a technique based on the Levenshtein distance [16]. Note that the Levenshtein distance is known nowadays mainly for its role in detection of plagiarism.

Any symbol of the string is constructed as a singular point, that is as a discontinuity of the first or second derivative of the range image; the alphabet of singular points to code a range image includes {STEP UP, STEP DOWN, ANGLE TOWARD ROBOT, ANGLE OUTWARD ROBOT} with obvious intuitive meaning for each option. Since this alphabet is rather short for reliable comparison of the strings, each symbol is supplied with an extended descriptor to reduce the probability of errors while taking a decision on occurrence of a HC.

Computation of the Levenshtein distance for two strings leads to so called Editorial Prescription (EP) that describes a simplest transformation of one string into another by means of some elementary operations over strings. If, for example, EP is empty then the strings are equivalent and there is no HC

between corresponding views; otherwise a decision on occurrence of HC is taken by a deeper analysis of EP. We refer the readers to [14] for further details.

Note that this approach cannot be applied directly to 2D images because the technique of Levenshtein is oriented exclusively toward processing the strings, so in general the detection of HCs is an open field for future research.

3 Inner Model of Environment

We start with some preliminary comments on the kind of robot sensor.

In experiments we deal with 1D range images, however the term ‘views’ does not refer to this particular kind of sensor. Note that, although different kinds of sensors generate views that seem qualitatively different, sometimes their information impact is almost equivalent. For instance, conventional video snapshots like in Fig. 1b can be chosen as an input flow and although such input seems rather different from the range images, the extraction of the range data virtually equivalent to that represented in Fig. 1c can be easily performed by processing the colour views as the robot moves in its workspace.

It is quite reasonable not using crude but somehow post-processed views to make them more robust about noise, illumination, etc., as well as more compact for storing. For example, instead of the crude range images similar to Fig. 1c, the strings described in section 2.3 are stored in our experiments.

The network of reference views mentioned in section 1 represents an inner robot model of its environment. It is constructed as a set of robocentric views specifically organized to meet navigation needs. The inner model comprises two databases: VIEWS that stores the set of reference views and ATLAS that imposes a kind of topology on VIEWS and represents them in the form of a set of neighbourhoods.

Each neighbourhood comprises three sets: the set $\{b_1, b_2, \dots, b_n\}$, where each b_i can be regarded as ID of a Reference View (RV) in VIEWS, the set of entrances, and the set of exits.

The entrances and exits have the same format of pair $\{\text{neighbourhood_ID}, \text{RV_ID}\}$, where the IDs refer to respective objects in ATLAS and VIEWS. For an entrance, values of the components specify such objects *from which* the robot can be moved to the Current View (CV) by a command, whereas for an exit they stand for the objects *to which* the robot can be moved starting from a view b_i that belongs to the neighbourhood that owns the exit.

In its turn, the correspondence of a robot position to a neighbourhood of ATLAS is the matter of the self-localization process mentioned in Section 1. Mathematically, self-localization for non-coordinate navigation can be regarded as correspondence: $\text{CV} \rightarrow \{\text{CN}, \text{CRV}\}$.

All transformations of the inner model are carried out by a logic triggered just at the moments when this correspondence fails to be established. The remainder of this section presents this logic.

In a general case, CV does not belong to VIEWS, and hence it may not belong to the set $\{b_1, b_2, \dots, b_n\}$ of CN. However for a correct self-localization in

a workspace, the reachability of CV from a reference view b_i is required. Under the assumption that the robot state in the CN yields view b_i the reachability means that, a single robot control command exists that moves robot from b_i to the state at which the CV has been produced, in other words, to the current state.

It is the task of the associative memory mentioned in Section 1 to construct an estimate of such command starting from a view b_i and CV. If the norm of the estimate is greater than a threshold, such estimate is rejected. If the same is true for all $b_i \in \{b_1, b_2, \dots, b_n\}$, the hypothesis of reachability of CV from CN is rejected as well.

In the process of learning the inner robot model, supervisor randomly generates a succession of robot control commands. Robot executes each command and gets a new CV; the new CV is subjected to analysis that leads to actualization of self-localization pair $\{CN; CRV\}$, as well as to possible modifications of both databases, ATLAS and VIEWS.

In this process, outcome of the following two tests completely determines possible modifications of the model: the Test on Reachability (TR) of CV and the Test on a Heavy Change (THC) between the previous view and CV.

A negative response of THC (no heavy change) means that CV corresponds to the same CN as the previous one, and the set of neighbourhoods of ATLAS remains the same. However, the result of TR is still important: VIEWS stays unchanged for a positive response, but otherwise the previous view of the input flow is added to VIEWS and included as a new element into set $\{b_1, b_2, \dots, b_n\}$ of CN. Since CV has been just reached from the previous view by the current robot command, this inclusion automatically provides the required reachability, and the just added view becomes a new CRV.

For a positive response of THC (heavy change occurred) further logic follows the block-diagram of Fig. 2: it starts from a straightforward search of an exit among those already registered in CN. If an appropriate exit is found, then CN must be substituted with the neighbourhood that the exit leads to. Just as in the previous paragraph, actualization of CRV depends on the response of TR: a positive response of TR is automatically accompanied with new CRV, whereas for a negative response, CV is included into VIEWS and added to the set of views of the actualized CN and CV becomes a new CRV since a view is reachable from itself.

If the search of exits fails, still a chance remains that robot has completed a long loop and returned back to an old neighbourhood. So search of loop closure is executed: CV is subjected to THC and TR with respect to the views of old neighbourhoods. For closure, it is necessary to find an old view with a negative response from THC and a positive from TR. Unfortunately, this condition is not enough to avoid errors completely: similar views can be perceived as distant states of robot. To reduce the probability of these errors, too uniform views can be filtered out from the search as unreliable; next, the sensor system of the robot can be enriched by a virtual compass (a sensor of orientation), which allows filtering out the candidates with a strongly different orientation than a

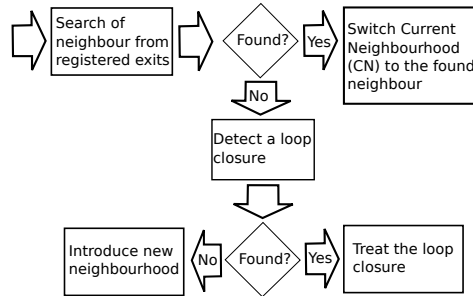


Fig. 2. Block-diagram of transformation of the model after a heavy change.

current robot orientation. As an ultimate resource, robot can call supervisor for help in doubtful cases.

If the search for loop closure fails, a new neighbourhood is introduced into ATLAS and the CV becomes its unique view. A link is established between the new neighbourhood and CN that is an entrance and exit are introduced into respective neighbourhoods; and then CN is actualized to be the new neighbourhood.

Otherwise, if a loop is detected it is treated as follows. First, it is clear that the CN turns out to be a new entrance to an Old Neighbourhood (ON), on which the loop closure has just been detected. Hence a corresponding link – new entrance to ON, new exit to CN – is introduced, but it is possible that this entrance is not brand new because it had been introduced earlier to ON under a different ID. To study this possibility, a view b_i from CN is subjected successively to THC with respect to the views of all entrances to ON. If an entrance E is found, whose view component yields a negative response of THC, the hypothesis is accepted that the neighbourhood component of E is identical to CN is accepted. Since the views under consideration are topologically close each other, although the errors of identification of views are still possible, their probability is much less than for loop closure. As a consequence, CN and the neighbourhood indicated in E can be merged into a single neighbourhood; depending on which neighbourhood is chosen to absorb the other, one of the two neighbourhoods can be deleted from ATLAS after merging. It also implies substitution of all references in ATLAS to the deleted neighbourhood by the ones to the remaining neighbourhood.

The previous paragraph, in a simplified form, presents the idea of a surgery of the inner model after a loop detection. In fact, a detected loop allows a deeper surgery to be applied. Indeed, THC can be applied, for instance, to the entrances of CN versus the entrances of entrances of E. Furthermore, the same idea can be extended to the exits. It would help to establish more equivalent neighbourhoods additionally to those of the previous paragraph.

Applying a deeper surgery makes the inner model to converge faster, but even the simple surgery leads to the convergence of ATLAS: the model tends to stabilize itself as the robot route becomes sufficiently dense in the workspace.

4 Experiments

A newer version of the pilot software described in [14] was used for the experiments. The functions implemented in the package include:

- Interactive creation of obstacles of the robot environment.
- Visualization of the environment and different kinds of robot views.
- Intuitive supervisor graphic interface for controlling robot.
- Training the Kohonen associative memory.
- Transformation of angle-distance plot to a string of singular points.
- Application of a technique based on the Levenshtein distance to the strings of singular points for detection of a heavy change.
- Auto-localization process.
- Manipulations with the inner model of the environment.
- Serialization/Deserialization.

Using software comprises three stages: generation of obstacles inside the workspace; training the associative memory and learning geography of the workspace.

For a generated workspace, just after the Kohonen associative memory has been trained, the program automatically initializes: the robot position shown by the triangle in Fig. 1a; the content of ATLAS of a unique neighbourhood and the content of VIEWS to a unique view like the one in Fig. 1c.

Supervisor drives robot so that each mouse click is interpreted as a leading point to be translated in a combination of rotation $< 4^\circ$ and displacement ahead $< 3\%$ of the workspace size. About 80 commands are required to close a loop shown by dashed line in Fig. 1a. The processing of each command of the sequence follows the logic presented in Section 3.

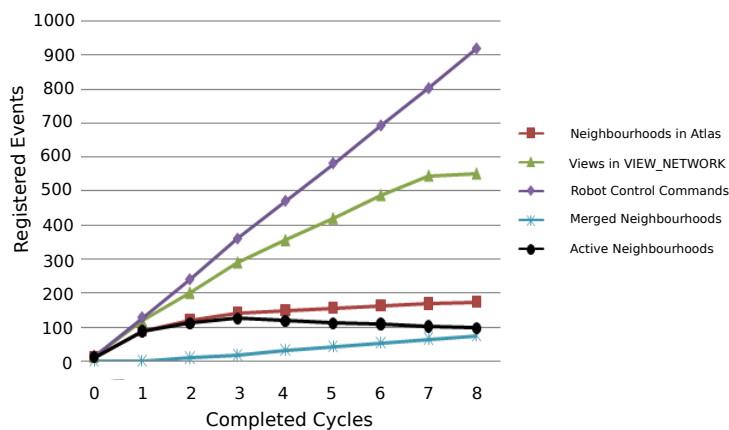


Fig. 3. Plot of statistics obtained during experiments.

A typical experimental result of learning the workspace is illustrated by Fig. 3. It presents several statistics that show the dynamics of generation of the inner model. The statistics are registered for a looped route that approximately follows the dashed line of Fig. 1a. Supervisor leads the robot several times along the loop, and the statistics are monitored as functions of the number of full revolutions.

The plotted statistics include the number of: (i) commands; (ii) neighbourhoods introduced into ATLAS; (iii) merged neighbourhoods; (iv) views introduced into VIEWS. One additional plot shows the number of active neighbourhoods in ATLAS defined as ‘introduced neighbourhoods minus merged neighbourhoods’ that is the difference (ii)–(iii). This plot shows an obvious tendency for stabilization of the fragment responsible for the loop in question in the model under construction.

5 Conclusions

The presented approach for a robot learning from its workspace employs a graph-based model ideologically similar to the view graph, [8, 18], or the graph G of the GT-model, [13]. The inner model of workspace is constructed as a set of reference views from the input flow, organized in an atlas of neighbourhoods. Our approach is parallel to SLAM, but no coordinate transformation is involved and no map of the workspace is build: the intention to avoid any coordinate representation is the main bio-inspired motivation for this work.

Instead of a explicit use of coordinate transformation we deal with its implicit substitute based on a local inversion of the cause-effect correspondence between a robot control and a sensor response; to provide this inversion, a Kohonen associative memory is used.

This work focuses on the logic of transformation of the inner model: adding new reference views; agglomeration them as neighbourhoods; switching a current neighbourhood; adding a new neighbourhood into the atlas and merging some neighbourhoods detecting a loop closure on a robot trajectory. Two tests govern learning the inner model: test on reachability of a current view from a reference view, and test on a semantic similarity of two views. The former decides whether a control exists that would provide a needed change of view; the latter determines if robot still belongs to a current neighbourhood. Only similar reference views (no ‘heavy change’ between them) constitute a neighbourhood.

The logic of creation of the inner model was tested on a virtual robot in a 2D workspace. The flow of input views in this case is represented by 1D-range images. For this kind of virtual sensor the detection of heavy changes can be effectively done, [14], by transforming each view to a string of singular points and a subsequent comparison of such strings by a technique based on application of the Levenshtein distance. The experiments carried out for this work give a statistical confirmation of stabilization of the inner model of the workspace when supervisor leads the robot many times along a cyclic trajectory.

This work proposes a novel method for automatic construction of a graph-based model of workspace, formed by reference views. Using such models opens

the way to solve navigation problems in natural terms. It can be done, [13], by introducing an additional structure to represent the ordinary geographic concepts. This issue presents a future research lines. Other lines for future works are: replacement of the 1D robot sensor with other kinds of sensors; extension of the technique from the 1D to 2D views, as well from a static to non-static workspace, and more. As a challenging goal oriented to practice, we can mention the development of a drone able to navigate automatically in a region after learning it just by flying several times over it.

References

1. Boniardi, F., Valada, A., Burgard, W., Tipaldi, G.D.: Autonomous indoor robot navigation using a sketch interface for drawing maps and routes. In: Robotics and Automation (ICRA), 2016 IEEE International Conference on. pp. 2896–2901. IEEE (2016)
2. Boularias, A., Duvallet, F., Oh, J., Stentz, A.: Grounding spatial relations for outdoor robot navigation. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. pp. 1976–1982. IEEE (2015)
3. Castellanos, J., Martinez-Cantin, R., Tardós, J.D., Neira, J.: Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and autonomous systems* 55(1), 21–29 (2007)
4. Cheeseman, P., Smith, R., Self, M.: A stochastic map for uncertain spatial relationships. In: 4th International Symposium on Robotic Research. pp. 467–474 (1987)
5. Davison, A.J., Murray, D.W.: Simultaneous localization and map-building using active vision. *IEEE transactions on pattern analysis and machine intelligence* 24(7), 865–880 (2002)
6. Dissanayake, M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation* 17(3), 229–241 (2001)
7. Eade, E., Drummond, T.: Unified loop closing and recovery for real time monocular slam. In: BMVC. vol. 13, p. 136. Citeseer (2008)
8. Franz, M.O., Schölkopf, B., Mallot, H.A., Bühlhoff, H.H.: Learning view graphs for robot navigation. *Autonomous robots* 5(1), 111–125 (1998)
9. Gomi, T.: Non-cartesian robotics. *Robotics and Autonomous Systems* 18(1-2), 169–184 (1996)
10. Grisetti, G., Tipaldi, G.D., Stachniss, C., Burgard, W., Nardi, D.: Fast and accurate SLAM with Rao–Blackwellized particle filters. *Robotics and Autonomous Systems* 55(1), 30–38 (2007)
11. Guivant, J., Nebot, E., Baiker, S.: Autonomous navigation and map building using laser range sensors in outdoor applications. *Journal of robotic systems* 17(10), 565–583 (2000)
12. Kähler, O., Prisacariu, V.A., Murray, D.W.: Real-time large-scale dense 3d reconstruction with loop closure. In: European Conference on Computer Vision. pp. 500–516. Springer (2016)
13. Khachaturov, G.: An approach to trip-and route-planning problems. *Cybernetics and Systems* 33(1), 43–67 (2002)

14. Khachaturov, G., Figueroa, J., González, S., Martínez, J.M.: Heavy changes in the input flow for learning geography of a robot environment. In: *Lecture Notes in Artificial Intelligence*. pp. 518–529. Springer (2017)
15. Kohonen, T., Lehtiö, P., Rovamo, J., Hyvärinen, J., Bry, K., Vainio, L.: A principle of neural associative memory. *Neuroscience* 2(6), 1065–1076 (1977)
16. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: *Soviet physics doklady*. pp. 707–710 (1966)
17. Mei, C., Sibley, G., Cummins, M., Newman, P., Reid, I.: Rslam: A system for large-scale mapping in constant-time using stereo. *International journal of computer vision* 94(2), 198–214 (2011)
18. Schölkopf, B., Mallot, H.A.: View-based cognitive mapping and path planning. *Adaptive Behavior* 3(3), 311–348 (1995)
19. Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S.J., Roy, N.: Understanding natural language commands for robotic navigation and mobile manipulation. In: *AAAI*. vol. 1, p. 2 (2011)

Image Encryption System Based on Cellular Automata and S-Box

Juan Contreras¹, Marco Ramírez¹, Jesús Aboytes²

¹ Coordinación Académica Región Altiplano Oeste-UASLP, Salinas, Mexico

² Instituto de Investigación en Comunicación Óptica, UASLP, San Luis Potosí, Mexico
juanjosetorres96@outlook.com, tulio.torres@uaslp.mx,
agustin.aboytes@upslp.edu.mx

Abstract. This investigation presents an image encryption system and its security analysis, it is based on cellular automata and a substitution box. The joint of these two techniques, allow to encrypt digital images with a high adjacent redundancy and pass different statistical and differential tests and cryptanalytic attacks. The synchronization phenomenon of cellular automata, is sensitive to initial conditions, therefore has been used in Pseudo Random Number Generators (PRNG) and cryptosystems. In this system, the synchronization phenomenon is used to change the coefficients of the pixels, but the process is not equal for each bit, thus the s-box is used to make the system robust and for compliance with the bit independence criterion among others.

Keywords: image encryption, cellular automata, S-box.

1 Introduction

Nowadays, we can perform many operations for internet, thus facilitating processes and optimizing times. But, this requires providing security to users, given that their data are exposed on the transmissions or at the storage location. One of the techniques used to protect information are cryptographic algorithms. This technique consists of making the information unintelligible, in such a way that it can only be recovered using the correct key.

Currently, image encryption is a very active field of research, due to the multiple areas where it is required, for example: in the pay-television service, medical imaging systems, videoconferences, military communications, video surveillance, among others. Although there are several conventional encryption algorithms, such as AES (Advanced Encryption Standard), DES (Data Encryption Standard) and IDEA (International Data Encryption Algorithm), they have often been impractical for image encryption, due to the intrinsic properties of these, such as large volumes, a strong adjacent correlation, a high redundancy, among others [1]. Therefore, the security problem extends because the algorithms for image encryption must provide perceptual security and cryptographic security.

This has encouraged the search and implementation of new schemes for image encryption, such as the large number of encryption systems with a chaotic approach [2, 3, 4]. That is why in this research the synchronization of cellular automata is combined

based on rule 90, which is of discrete chaotic dynamics and the substitution boxes (S-box) to provide a strong algorithm against cryptanalytic and statistical attacks and pass the visual inspection.

2 Background

2.1 Cellular Automata

The concept of cellular automata (CA) was introduced in the decade of the 40's by the mathematician John von Neumann and Stanislaw Ulam [5]. The CAs are used to model complex behaviors where local interactions are involved. In fact, CAs represent a class of dynamic systems capable of describing the evolution of systems using simple rules, without the need to use differential equations.

The cellular automata consist of an ordered set of cells, in the form of a grid, where each cell has a finite number of states. The cellular automata form a grid of two dimensions, where their cells evolve in discrete steps according to a local rule of update applied uniformly, on all cells. At the beginning, a state is assigned to the cells at time $t = 0$, where the new cell states will depend on their previous states and those of their neighborhood, see Fig 1.

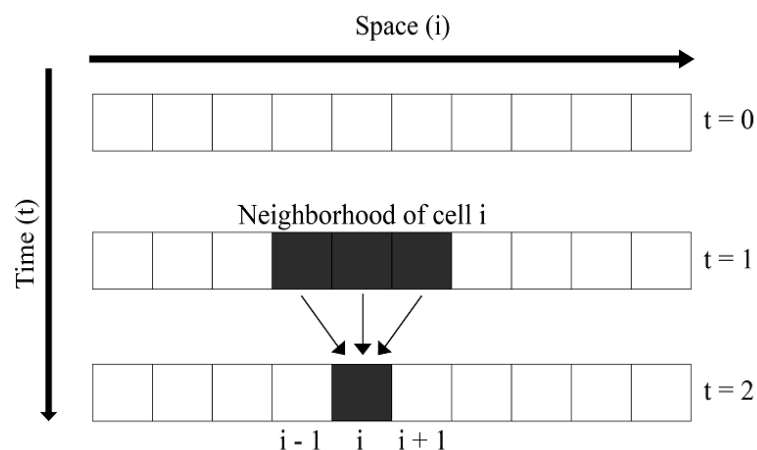


Fig. 1. Space-time diagram of a one-dimensional cellular automaton.

The elementary cellular automata (ECA) are CA of one dimension, with two states and neighborhood of radius 1. A local rule of cellular automata is the algorithm used to calculate the next state of the cell. The ECAs differ among themselves, only by choosing the local rule, they contain only three variables (cells) and each one can take only two values (1,0), therefore there are only 8 combinations, resulting in $2^8 = 256$ local rules and ECA different. For example, the local rule 90 is described by the following expression:

$$x_i^{t+1} = A(x_{i-1}^t + x_{i+1}^t). \quad (1)$$

The phenomenon of synchronization occurs when, after a period of time, the behavior of two dynamic systems approaches arbitrarily. In the case of CA, after a number of steps in time t , the difference between the vectors \mathbf{x} and \mathbf{y} corresponding to the controller and replicating cellular automaton respectively, will eventually result in the zero vector $\mathbf{0} = (0,0 \dots ,0)$. For this it is necessary that in each step, both vectors evolve using the same local rule.

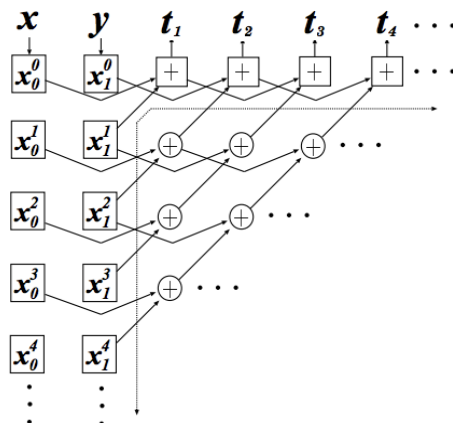


Fig. 2. Generator of pseudo-random sequences.

In Ref. [6] it was shown that a pair of ECAs that evolve using the local rule 90, synchronize if the coupled coordinates are separated by a block of $N = 2^n - 1$ decoupled sites, where n is a positive integer. Based on the phenomenon of synchronization, in Ref. [7] the authors proposed a Pseudo-Random Number Generator (PRNG). The main function is called h , and it requires two vectors \mathbf{x} and \mathbf{y} of n bits and $n + 1$ bits respectively. To calculate a pseudo-random sequence, the function requires that the cellular automaton evolve backwards. Such a situation is described in Fig. 2.

Where the XOR gates are represented with the circles that in the middle have a cross, the connectivity of these represent the local rule 90, and the resulting vector is called vector t .

In Ref. [8] a preprocessing function was created to exchange the values of the plaintext, based on the pseudo-random number generator, making a modification in its feedback, see Fig. 3. The process applied to images consists in receive each pixel coefficient as if it were the vector \mathbf{x} , the vector \mathbf{y} will be replaced after each iteration by the resulting vector $\hat{\mathbf{m}}$, concatenating the least significant bit of the vector \mathbf{y} precedent as the most significant bit of the new vector. This function allows to break the high adjacent correlation of the images, allowing a dynamic substitution of the information.

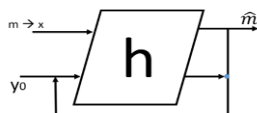


Fig. 3. Preprocessing function based on the h function.

2.2 S-box

On the other hand, in cryptography, substitution boxes are a basic component in symmetric algorithms. The boxes are used in cipher blocks to exchange the plaintext and in this way hide the relationship between the encryption key and the encrypted text [9].

The design and selection of a replacement box is a careful process, because it requires to be resistant to attacks of cryptanalysis. See Fig. 4 shows the S-box used in the AES encryption system, it has been tested and pass all the design principles.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	IE	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Fig. 4. S-box of the AES system in hexadecimal notation

3 Encryption Algorithm

3.1 New *h* Function

For the development of this image encryption algorithm, it is feasible to join both tools, synchronization phenomenon of CA and substitution boxes. The preprocessing operation is able to break the high correlation of the images and the substitution boxes provide security against attacks of differential cryptanalysis and bit independence criterion.

To increase the initial condition, a new version of the preprocessing operation was performed, where three *h* operations are used. So that there is no correlation between the original image and its processed version, see Fig. 5.

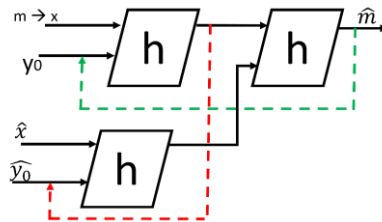


Fig. 5. Improved preprocessing function with three h functions.

3.2 Description of the Algorithm

The algorithm to encrypt an image works in the following way:

- 1° Select a block of plaintext 24-bits long (3 pixels in gray scale).
- 2° The new preprocessing function is applied to the plaintext blocks of the image.
- 3° The preprocessed block is divided in blocks of 8-bits, the value of each block is replaced using the S-box.
- 4° Subsequently, the columns and the rows of the resulting image are inverted in such a way that the pixel (n, n) now occupies the place $(0,0)$, calculating the complement of each coordinate.
- 5° Finally, the preprocessing function with the transformed image is used again.

The secret key of this algorithm is at least 148 bits since two extended preprocessing functions are used, therefore, given the current processing is not susceptible to breaking using brute force attacks [10].

4 Results

The results of the security analysis applied to the encrypted images are shown in this section. Various statistical tests, cryptanalytics attacks and the calculation of NPCR (Number of Changing Pixel Rate) and UACI (Unified Averaged Changed Intensity) indices were applied to validate the quality of the encryption. For the tests, we use images widely used in the image processing with different characteristics: mandrill, Lena and peppers. All in gray scale to 8 bits and dimensions 512 x 512 pixels.

4.1 Histograms

The first test consists of calculating histograms of both the plainimage and its encrypted version, see Fig. 6. It shows the case of the Lena image, where we can see that the histogram of its encrypted version is uniform, thus hiding the redundancy of data from the original image.

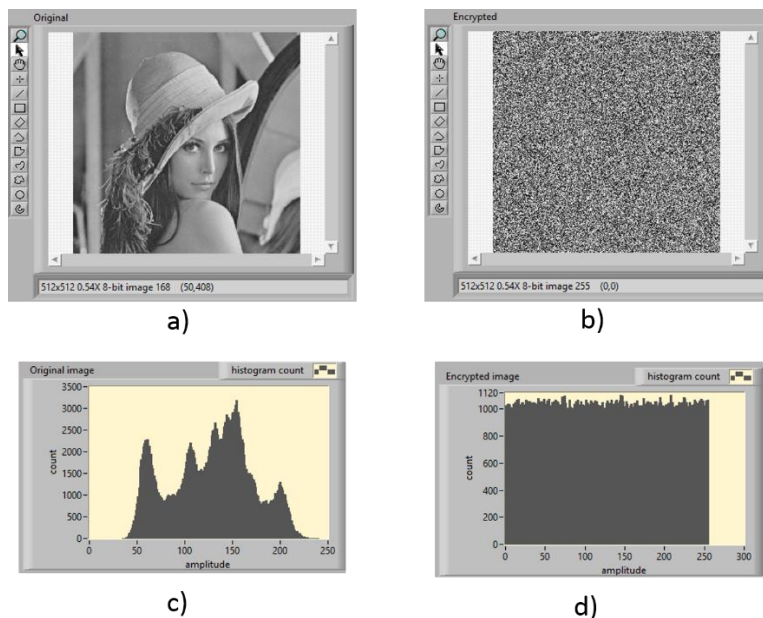


Fig. 6. Analysis of histograms a) Image of Lena, b) Image of Lena encrypted, c) Histogram of the original image of Lena and d) Histogram of the encrypted version of Lena.

4.2 Correlation

The second test that was performed was the calculation of the correlation coefficient between the original image and its encrypted version. This test tries to demonstrate the independence that exists between the encrypted image and the original image. According to the interpretation of this coefficient, there is no correlation between the images if the result is close to 0. Table 1 shows the results of this test applied to the three test images.

Table 1. Correlation coefficients between plainimages and their corresponding encrypted version.

Image	Coefficient
Peppers	-0.0025700
Lena	0.0011410
Mandrill	0.0005676

4.3 NPCR and UACI

In the encryption of images, it is common to analyze the resistance of the algorithms to differential attacks using two measurements: NPCR and UACI. Both measurements are based on small changes in two images and encrypt them under the same key. To illustrate this, let us assume that we have two encrypted images C^1 and C^2 , whose corresponding plainimages have only one pixel different from each other, and both have been encrypted with the same secret key. The coefficients in the grayscale of both

images in row i and column j are marked as $C^1(i, j)$ and $C^2(i, j)$ respectively. The NPCR and UACI indices are defined in equations (2) and (3).

$$NPCR: N(C^1, C^2) = \sum_{i,j} \frac{D(i, j)}{T} \times 100\% \quad (2)$$

$$UACI: U(C^1, C^2) = \sum_{i,j} \frac{|C^1(i, j) - C^2(i, j)|}{F \cdot T} \times 100\% \quad (3)$$

where $D(i, j)$ is determined in the following way: if $C^1(i, j) = C^2(i, j)$, then $D(i, j) = 0$, otherwise $D(i, j) = 1$, T is the total of pixels of the images and F denotes the maximum value valid in the format of the image. For gray scale images at 256 levels, the theoretical values are $UACI = 33.464\%$ and $NPCR = 99.609\%$, [11]. It is important highlight that to demonstrate the operation, a pixel will be modified to the image. The results obtained for our algorithm are shown in Table 2, where the least significant bit of the pixel $m(255,255)$ was modified.

Table 2. Numerical results for NPCR and UACI.

Image	NPCR	UACI
Peppers	99.6235%	33.434992%
Lena	99.6021%	33.425587%
Mandrill	99.6128%	33.361058%

As is shown in the Table 2, the results in the most cases are upper than the theoretical values, just in one case is lower for 0.0069%.

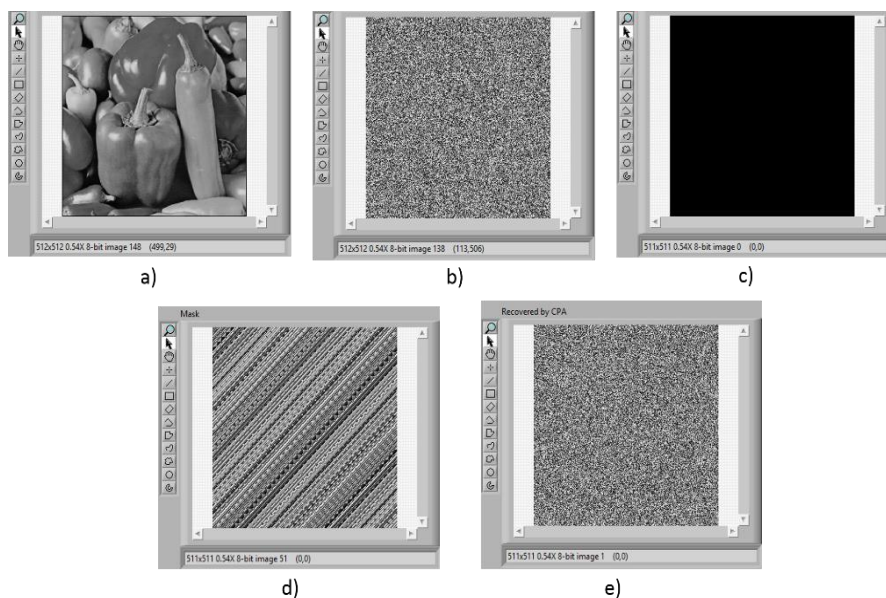


Fig. 7. Chosen-plainimage attack applied to the test image of the peppers. a) Original image, b) encrypted image of the peppers, c) solid image chosen, d) image mask and e) the recovered image.

4.4 Chosen-plainimage Attack (CPIA)

Finally, we performed the Chosen-plainimage attack (CPIA). In Ref. [12] they point out that if a cryptosystem is secure against CPIA attack, it is also safe against other cryptanalysis attacks such as cipherimage-only attack or known-plainimage attack.

This attack implies that the adversary is able to choose the plainimages and obtain their respective encrypted version, but he does not know the secret key. The attack begins by selecting the images to be encrypted, as can be seen in Fig. 7, the image of the peppers is used, Fig. 7a) and a solid black image, Fig. 7c). Both images are encrypted under the same secret key, the results are Fig. 7b) and 7d). Finally, an XOR operation is performed pixel by pixel between both encrypted images, the result will be what is called the recovered image, Fig. 7e). As we can see in our case the resulting image does not reveal information of the original image.

5 Conclusions

In the present work, a new algorithm for image encryption was proposed, the quality of encryption was measured with different tests and kinds of images, offering cryptographic and perceptual security, in all the cases. Both tools, the synchronization of cellular automata and the substitution boxes are complemented to securely encrypt this information. This encryption system could be a feasible option to encrypt high redundancy image, without reveal any pattern. Therefore, it presents a better performance than AES system in ECB (Electronic Code Book) mode.

References

1. Lian, S.: Multimedia content encryption: techniques and applications. Auerbach Publications (2008)
2. Niyat, A.Y., Moattar, M.H., Torshiz, M.N.: Color image encryption based on hybrid hyperchaotic system and cellular automata. *Optics and Lasers in Engineering*, 90, 225–237 (2017)
3. Belazi, A., El-Latif, A.A.A., Belghith, S.: A novel image encryption scheme based on substitution-permutation network and chaos. *Signal Processing*, 128, 155–170 (2016)
4. Belazi, A., El-Latif, A.A.A., Diaconu, A.V., Rhouma, R., Belghith, S.: Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. *Optics and Lasers in Engineering*, 88, 37–50 (2017)
5. Von Neumann, J., Burks, A.W.: Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, 5(1), 3–14 (1966)
6. Urias, J., Salazar, G., Ugalde, E.: Synchronization of cellular automaton pairs. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 8(4), 814–818 (1998)
7. Urias, J., Ugalde, E., Salazar, G.: A cryptosystem based on cellular automata. *Chaos*, 8(4), 819–822 (1998)
8. Ramirez-Torres, M.T., Murguia, J.S., Carlos, M.M.: Image encryption with an improved cryptosystem based on a matrix approach. *International Journal of Modern Physics C*, 25(10), 1450054 (2014)
9. Chandrasekaran, J., Subramanyan, B., Selvanayagam, R.: A chaos based approach for improving non linearity in S box design of symmetric key cryptosystems. In: *International Conference on Computer Science and Information Technology*, pp. 516–522. Springer, Berlin, Heidelberg (2011)

10. Paar, C., Pelzl, J.: *Understanding: a textbook for students and practitioners*. Springer Science & Business Media (2009)
11. Wu, Y., Noonan, J.P., Agaian, S.: NPCR and UACI randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)* 1(2), 31–38 (2011)
12. del Rey, A.M., Sánchez, G.R., De La Villa Cuenca, A.: Encrypting digital images using cellular automata. In: *International Conference on Hybrid Artificial Intelligence Systems*, pp. 78–88. Springer, Berlin, Heidelberg (2012)

MeIPeAS: An Intelligent Virtual Tutor for Mexican Elementary Schoolchildren

Hector Rafael Orozco Aguirre, Gonzalo Ivan Riego Caravantes

University Center UAEM Valley of Mexico,
Autonomous University of Mexico State, Mexico
hrorozcoa@uaemex.mx, vannlink@hotmail.com

Abstract. It is known that virtual tutors have a wide range of functionalities, which have been little exploited and applied in the educational field at the primary level. However, these functionalities allow to offer mechanisms of interaction with students through an interactive dialogue by using text to speech, and even more sophisticated, the recognition and understanding of natural language or speech. In this paper, a personalized virtual tutor for the primary education scenario in Mexico is presented. This virtual tutor is called Mexican Intelligent Pedagogical Agent for Schoolchildren (MeIPeAS) and was created to be used as a pedagogical support mechanism offering a unique attraction for current and future generations of schoolchildren in Mexico. The virtual tutor has been validated in practice in public primary schools of the municipalities of the State of Mexico in Mexico. This validation is to analyze the impact of the user experience from the obtained results having relevant information about the reinforcement of topics taught within the classroom.

Keywords: education, intelligent agent, pedagogical agent, virtual dialogue, virtual tutor.

1 Introduction

Interactive agents used for pedagogical purposes have demonstrated to induce a higher motivation to learn on students. These agents known as pedagogical agents would enhance learning when attracting attention through a dynamic interactive and conversational mechanism with students by a bidirectional dialog. Pedagogical agents look like anthropomorphic virtual characters used primarily for educational or training purposes. The design of pedagogical agents has changed over time depending on the desired objectives through their use. This field of research has emerged over the past three decades to provide support and achieve an adequate teaching-learning process according to the student and teacher needs. Apart of being able to communicate in a verbal way, it is desirable that created pedagogical agents are full bodied, and can show emotions or facial expressions even body motions in order to be able to communicate in a non-verbal manner with students. There exist different ways to animate and represent pedagogical agents, some of the most common are: using a cartoon-style character, playing real videos or employing 2D or 3D models.

Pedagogical agents, which are able to show affective reactions, have expressive behaviors, exhibit a personality, speak and listen to the user. These can offer a further

dimension to the experience of human-computer interaction applications for learning purposes. Their construction makes possible to extend the experience within the learning environment by enhancing communicative setting, delivering a capability where the spoken output can be enhanced by expressive talking agents that can understand student speech and are capable to speech to them. The point of departure for the approach described in this paper is that this agent technology can be used in the development of an effective pedagogical agent in the form of an intelligent virtual tutor for supporting and assisting the teaching-learning process within the first three grades of primary education in Mexico.

Given the above, in this paper a virtual tutor prototype called Mexican Intelligent Pedagogical Agent for Schoolchildren (MeIPeAS) is presented, as a new tool based on human-computer interaction techniques through the approaches of serious games [1] [2] and educational software [3] [4]. That is, so that students into the first three grades of primary education in Mexico can interact with it, while seeking to achieve understanding and master the acquired knowledge in each theme of each block included in all subjects that must be studied.

The interactive virtual tutor here presented is intended to support the teaching-learning process within the classroom, not as a substitute for a real tutor or teacher, but as a pedagogical support element even at home. This virtual tutor represents a new working tool that can be used in teaching, and helps in the increase and remarkable improvement of the educational quality from primary education levels, being one of the first pedagogical agents developed in Mexico. This agent inhabits a serious game acting as an individualized teaching-learning platform and intelligent tutoring system. With the use of the virtual tutor (see fig. 1), it will be verified that it is feasible to provide a supporting tool for the improvement of educational didactics, as well as a reinforcement of the subjects taught in the first three grades of primary education in Mexico, seeking to demonstrate that means of reinforcement exercises for the subjects included in the current curriculum, a simple, creative, dynamic, intuitive and fun interaction mechanism is achieved, which assists and supports the teaching-learning process by impacting the user experience in a positive manner.

2 Pedagogical Agents and their Applications

Animated agents are graphical representations of characters used in computer interfaces where a human-computer interaction in a more personalized way and natural style is preferred to be present among users and agents. These virtual characters have been created for a wide variety of computer applications such as virtual presenters, training agents, and pedagogical agents. In pedagogical applications, users tend to prefer a learning system which employs an animated agent to a similar one without an agent. Animated pedagogical agents are often designed as human-like characters that facilitate the learning process and support students in solving activities in learning platforms. These kind of agents is also a useful tool as an aid for teachers in schools during learning activities. The use of animated agents in pedagogical applications has been effective in intelligent tutoring systems in which students can improve the learning experience by engaging dialogue or conversation with agents, that is, a learning environment can be enhanced with the addition of animated agents. When a pedagogical agent is facially

animated within a learning platform students experience a more natural learning process because the agent positively affects student's perception of their own learning experience by its presence.



Fig. 1. Individual use of the presented pedagogical virtual tutor within a teaching-learning environment for Mexican schoolchildren.

In [5] cited by [6], authors detail types of interaction that agents can display, which benefit the learning context, and also described the following benefits that animated pedagogical agents can bring to human-computer interaction applications:

- Pedagogical agents are able to provide to students how to perform a particular task. They can virtually show how to complete a task into a virtual learning environment.
- Pedagogical agents can act as navigational guides to students in complex learning environments.
- Pedagogical agents can steer a student's attention to something in the virtual world by way of gaze behaviors and deictic gestures.
- Pedagogical agents can provide non-verbal feedback to a student's input or actions, as well as verbal feedback.
- Pedagogical agents are able to express conversational signals which people are accustomed to in human-human interaction, for turn taking, expressing personal opinions or acknowledging the user's utterance.
- Pedagogical agents can convey emotion to the user which in turn may elicit emotion from the user and serve to increase learning motivation.
- Pedagogical agents can serve the role of a virtual teammate, where working in a team is an element of the task design. Here, the agent can either act as an instructor, helping the student to accomplish a task, or substitute as a missing team member, allowing the student to practice working in a team.

According to [7], an agent can exist within the learning environment, but it is also possible for an agent to exist in a separate window. In addition, it is preferred that the behavior be generated ideally online and in a dynamic way instead of being created offline or manually. A dynamic behavior is better and corresponds to the changes into the learning environment.

Previous efforts on pedagogical agents include empirical researches about their uses, effectiveness, or limitations. The most representative examples of successful on agent-

based learning applications are summarized in [8]. Some of these pedagogical applications are the following:

- Hernan the Bug: the first agent developed to inhabit virtual environments to help middle school students to understand botanical anatomy and physiology.
- Soar Training Expert for Virtual Environments (STEVE): this was designed to interact with students in networked virtual environments and applied to naval training tasks.
- Cosmo: a life-like animated agent with deictic believability that occupies the Internet Adviser for helping learners on Internet packet routing.
- PPP Persona: it was created to guide the learner through web-based material using presentation acts to draw attention to elements of the web pages, and provide commentary via synthesized speech.
- JACK: a virtual human that exhibits a variety of different deictic gestures as a virtual presenter who can point at individual elements on his visual aid and integrates these gestures into his presentation, moving over to the target object before his speech reaches the need for the deictic gesture.
- ADELE: an agent designed to operate over the web able to extend the pedagogical capabilities of STEVE agent in a wider range of educational problems.
- VINCENT: a tutor friend for on-the-job training to assist the student, promote his confidence and motivate the student to learn.

Recent efforts on pedagogical agents show how agents and applications evolved in order to provide better learning platforms and solutions with less restrictions, where students expect agents to be believable as virtual mentors, be entertaining, easy to communicate, helpful, and diversified. In [8], there are also explained some of the most relevant agents, which are the next ones:

- Automated Lab Instructor (ALI): this agent provides flexible guidance to students interacting with virtual labs and had been applied to chemistry and biology simulations and also its application to virtual factory teaching system after that provided further evidence of its generality.
- MASCARET: a pedagogical multi-agent system for training proposed to organize the interactions between agents and to provide them reactive, cognitive and social abilities to simulate the physical and social environment.
- SKIP: a puppet agent used to improve science learning through the use of digital puppets in peer teaching and collaborative learning settings.
- Mediating Agent of MACES: an animated agent that has the role of providing emotional support and promoting in student a positive mood.
- MENTOR: a virtual teacher who monitors student activities and collaborates to other guide agents of environment. It allows users to travel to virtual places to perform educational activities, talk with other users and mentors, and build virtual personae.

Kim and Baylor [9] review what students want from a pedagogical agent it is a good teaching ability to be knowledgeable and the ability to give motivation being friendly

and kind when using various instructional strategies in an interactive learning environment. Students significantly learn better and have greater motivation when working with two agents (split persona effect) than with one mentor agent. Mixed design factors of pedagogical agent research significantly improve learning performance and student behavior. A good example is described in [10], where a virtual tutor called Ms. Readwrite gives hints and explanations to help children to figure out answers while learning about phonological awareness, reading, spelling, and comprehension. This virtual tutor helps children learn to recognize words, to read fluently, and to understand what they read, providing engaging practice with individualized support and focused hints. My Science Tutor (MyST) is described in [11], representing an intelligent tutoring system designed to improve science learning by students in 3rd, 4th, and 5th grades through conversational dialogs with a virtual science tutor. Individual students engage in spoken dialogs with the virtual tutor Marni during 15 to 20 minute sessions following classroom science investigations to discuss and extend concepts embedded in the investigations. The spoken dialogs in MyST are designed to scaffold learning by presenting open-ended questions accompanied by illustrations or animations related to the classroom investigations and the science concepts being learned.

As shown in [12], some of the motivation theories are used as part of the pedagogical agent design. The most frequently used in pedagogical agent research is the social agency theory as a theoretical framework, assuming that sounds and images in pedagogical agents build social cues and trigger social responses improving deeply learning in student activities. As a final argument, it is argued that a pedagogical agent should tend to be a teachable agent able to learn while teaching. Blair et al. [13] defined a teachable agent as a one that works by letting students teach them and then assessing their knowledge by providing a series of questions to solve problems. In [14], it is mentioned that learning-by-teaching pedagogy has been identified as one of the more effective approaches to learning, because in a face-to-face learning-by-teaching situation, the role of the learners is to teach their peers or instructors. In virtual environments, learners take an active role by teaching a computer agent, which is referred to as teachable agent. In this way, a virtual learning-by-teaching environment is presented. A communication method was adopted in this system. The method facilitates the interaction between the learner and the computer agent, specifically for K-12 students' mathematics learning.

According to [15], due to pedagogical agents are becoming more common and complex, it is important to develop a better understanding of how best to combine interactive agents with teachers and students, to complement and make the learning process most effective and positive. Pedagogical agents should not be seen as taking the place of teachers but as a supporting tool in the learning process within the classroom [16].

3 Mexican Education Scene

In Mexico, a good quality education [17] can be sustained if it contains the following attributes: relevance, pertinence, internal efficacy, external efficacy, quality, sufficiency, good educational results, and equity. However, within education

institutions at basic levels, despite the creation and implementation of new methodological teaching proposals towards mastery of competences, in practice there have been disadvantages, such as numerous groups, which is a cause that hinders a good start-up of an adequate teaching-learning process.

An integral reform of education in Mexico [18] in each of the levels that integrate basic education, must be focused on providing students with a learning scheme based on the development and application of competencies. This learning scheme has been involved in a lag due to the educational models and teaching-learning media that are applied [19], where in general; the theory is reviewed and explained using media and materials such as printed texts and digital or virtual resources. Students understand little information and acquire knowledge that they do not know how to put it into practice beyond the classroom. In this way, these resources have been very badly exploited, leading to failure to inculcate the bases for the development of basic scientific training and the acquisition of notions about information technology and communications.

Trying to remedy the problem above mentioned and as a point in favor, it is well known that the new information and communication technologies have been little used, incredible as it may seem, being partially or totally outside the educational processes, since they reach to improve the educational didactics. In addition, these technologies facilitate the achievement of a broad and better integral development of the students, as much in their abilities of learning as of application of knowledge from basic levels of education.

In Mexico, the use of computing applications within the educational didactic is very little applied, or failing that, they are not correctly exploited in several of the school levels, specifically in primary education. When developing computing applications in the educational field, it is intended that their use provide an aid in teaching into the classroom, or being a useful tool to assist learning at home. This fact, leads to use the term, educational software, which is based on learning models, as well as concepts of cognitive and constructivist psychology. On the other hand, there is a subcategory known as virtual educational tutors, on which this paper is based.

The last integral reform of basic education in Mexico has as main challenge to raise the quality of education in the country, through continuous improvement of technological resources and high training of teachers and teaching to students, so that both take advantage of innovation present in the use of new didactic resources into the classroom. This may be possible through the incorporation of information and communication technologies into the professional training and pedagogical processes [20], taking for example the following ones:

- The use of digital tools and resources to support the understanding of knowledge and concepts.
- To plan and manage research, using information and communication technologies.
- To use models and simulations to explore some topics.
- To generate original products with the use of information and communication technologies, in which critical thinking, creativity, or the solution of problems based on real life situations are used.
- To make responsible use of software and hardware, whether working individually, in pairs or as a team.

- To inculcate ethical, safe, and responsible use of Internet and digital tools as well.

Within the plans and programs of study of basic education in Mexico, the consolidation of a curriculum that encourages the development of competences in the use of information and communication technologies, is in response to the social demand and the quality of the Mexican public school [21]. To achieve this, it is known that the improvement of the infrastructure and equipment of the primary schools, in particular of their laboratories and workshops, is a priority. To do this, it must be based on digital technologies to equip students and teachers of devices and technological links.

It should be noted that in the first three grades of basic education, it is where the bases for the development of basic scientific training and the acquisition of notions about technology are established as mentioned in [22]. The context is clear, any educational reform should not evade digital skills standards, and through these it will be possible to teach the proper use of information and communication technologies, being the fundamental basis for the development and acquisition of skills [23].

4 Virtual Tutor for Mexican Schoolchildren

In an initial stage, there are some aspects to keep in mind while developing the virtual tutor into the Mexican education scene, which are the following ones:

- The benefited primary schools are some of those belonging to the municipalities of the State of Mexico in Mexico.
- The topics of the courses taken within the first three grades of primary education will be considered according to the current curriculum, which are: Spanish, Mathematics, Exploration of Nature and Society, Civic and Ethical Education, and finally, Artistic Education.
- Only some reinforcement exercises will be created, which will be taken as models to create the remaining ones in a final version of the virtual tutor.
- Students can choose the gender of the virtual tutor, as they feel better.
- The voice of the virtual tutor will be synthetic and will be reflected through a bidirectional dynamic dialogue, both to the student through text to speech and to the tutor through speech recognition.
- The virtual tutor will not be able to express emotions neither verbally nor visually. However, for a final future version this aspect will be fully covered.

4.1 Module Description

One of the most outstanding aspects in the design of the educational virtual tutor is the human-machine interaction, with an attractive design to capture the attention of the students, achieving an effective natural relationship in real time. The general outline of the prototype is shown in fig. 2, which is composed of three interconnected modules and multimedia components. The functionalities for each module are described as follows:

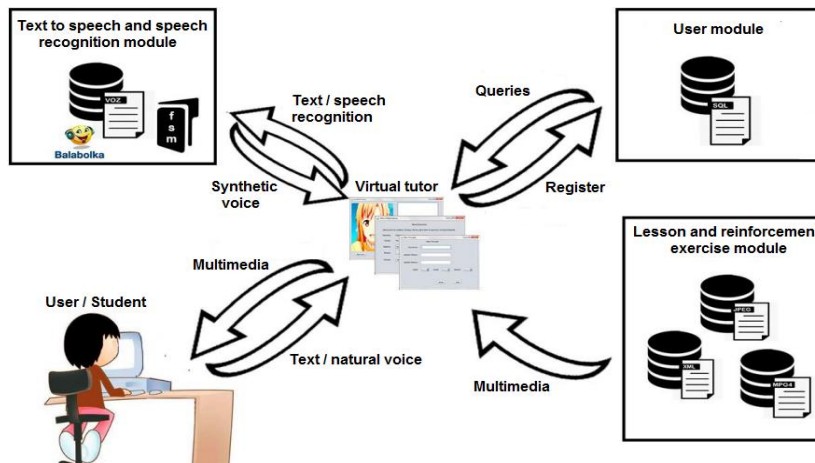


Fig. 2. General outline of the operation, design and implementation of the virtual tutor.

4.1.1 User Module

This module has a database where the basic data of the users will be registered, which are their names, paternal surname, maternal surname, age, grade, gender, password that will automatically generate the system, the school to which they belong, as well as the municipality where they are located. Thus, the virtual tutor will have information about each student to interact in a personalized and unique relationship.

4.1.2 Lesson and Reinforcement Exercise Module

This module is formed by a first database of images that are used in the reinforcement exercises and the possible personifications of the virtual tutor (female or male), with a caricature style; and a second database with explanatory videos on the topics that are addressed into the curriculum of the first three grades of primary education in Mexico; and finally, a third database with XML files based on the Java Speech API Markup Language, whose structure is in the form of a finite state machine, being considered for making decisions corresponding to the actions performed by students and the virtual tutor MeIPeAS.

4.1.3 Text to Speech and Speech Recognition Module

This module performs the action of converting the text introduced by the student through the keyboard or the options that he/she chooses to the personification of the virtual tutor's voice, that is, text-to-speech conversion. In addition, it is responsible for recognizing and processing the student's speech at the time the virtual tutor asks a question and waits for a verbal response. For the functioning of this module authorship code and open source solutions are used like the command line text to speech utilities voice.exe from Eli Fulkerson and speech.exe from Balabolka. However, MeIPeAS was created to also use commercial software such as Microsoft Speech Platform, CereProc and Amazon Speech Recognition Software.

4.2 Implementation as a Pedagogical Agent

To personify the virtual tutor (see fig. 3), GIF images of caricatured appearance were used, as well as the corresponding animation to represent the speech action according to the case of the genre. The audio and speech animation in each GIF image are synchronized to denote a better naturalness during the interaction with the student.

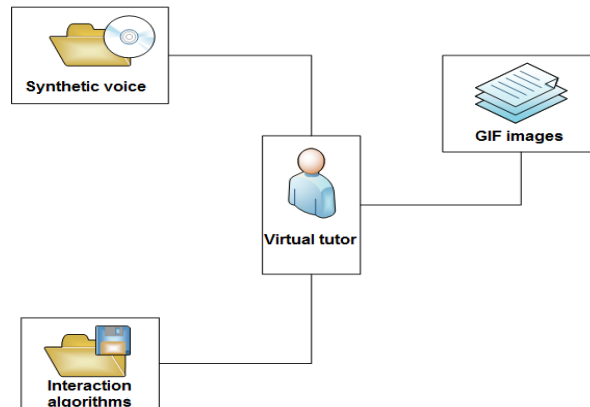


Fig. 3. Personification by means of synthetic voice and GIF image of the virtual tutor.

Once a student has started a session, he/she must choose in each subject the respective block and topic on which he/she wishes to obtain a review information and take the challenge of responding to a reinforcement exercise. This choice is made from the window shown in fig. 4. Fig. 5 shows the window used to start an exercise or well to exit to take another one from the window shown in fig. 4. The virtual tutor is representing by a female character.

As it is possible to notice, Spanish was used as the interface language given in the windows of the prototype as well as in the mechanism of interaction through the bidirectional dialogue with the virtual tutor. This is because in Mexico Spanish is the official language and the vast majority of schoolchildren who course the first three grades of primary education are having their first approaches with a second language such as English as a foreign language.

At the moment, only students can register and log in with the virtual tutor. However, it is intended that the teacher can do it equally but in such a way that he/she can have the options to add and modify lessons. A more ambitious challenge would be that this prototype could be available via the Internet, as well as be able to run on various platforms such as Android, IOS, Windows, and Linux systems.

Some primary schools from the municipalities of Atizapan de Zaragoza, Tlalnepantla de Baz and Nicolas Romero from the State of Mexico in Mexico were pilot testing scenarios. The prototype was presented to teachers and students from the first grade of basic education, as shown in fig. 6, doing the following:

- It was explained to the students and teachers about the prototype, the objectives that are sought to achieve, the benefits that this can have within the educational field and the way of interacting with the virtual tutor.



Fig. 4. Menu of exercises according for each subject.

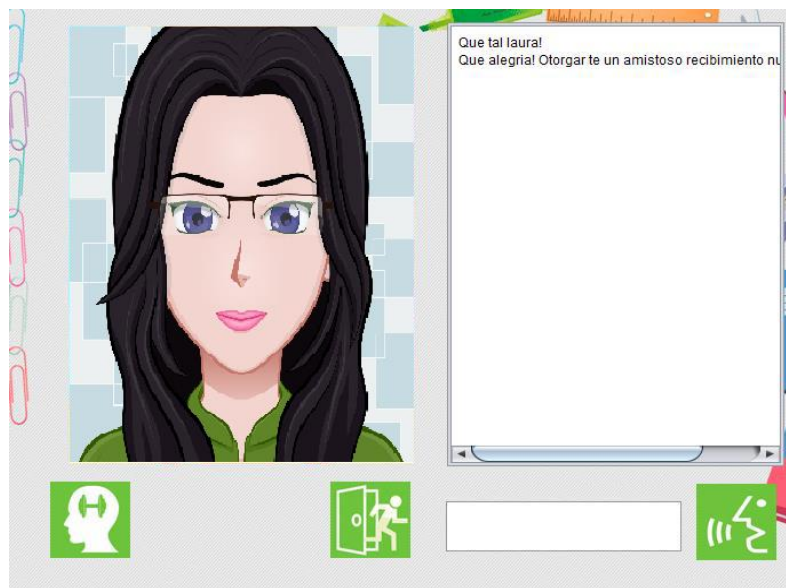


Fig. 5. Welcome to the student.

- A demonstration of the operation of the prototype was given, explaining from how it is executed, the content of each menu, the way they should register, how to log in and use the exercises, as well as the interaction with the virtual tutor.
- Once the explanation was completed, teams of students were formed to interact with the virtual tutor prototype.



Fig. 6. Testing interaction scenario between students and the virtual tutor.

An example of reinforcement exercise performed is given in fig. 7. It is observed that the student must identify the animal shown by its name, in this case a gorilla. Other example of reinforcement exercise is shown in fig. 8, where a student has to identify the corresponding image to a body part asked by the virtual tutor. The virtual tutor is representing by a male character.

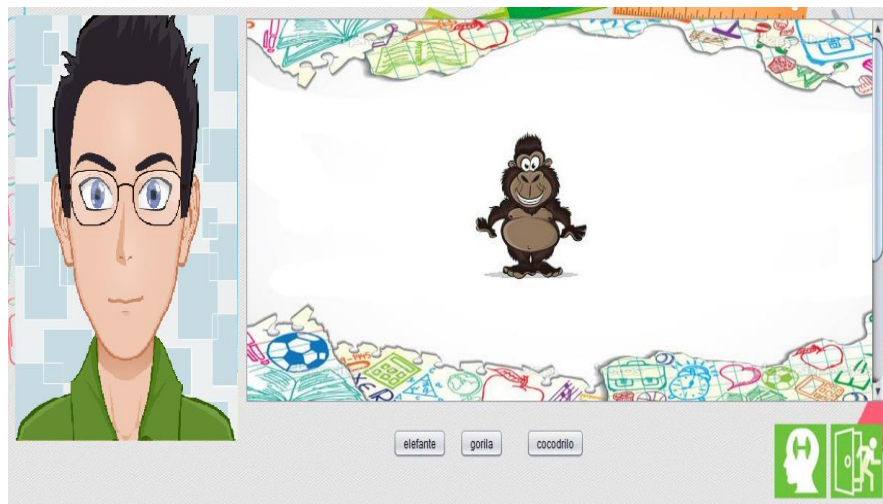


Fig. 7. Reinforcement exercise related to a Spanish course.

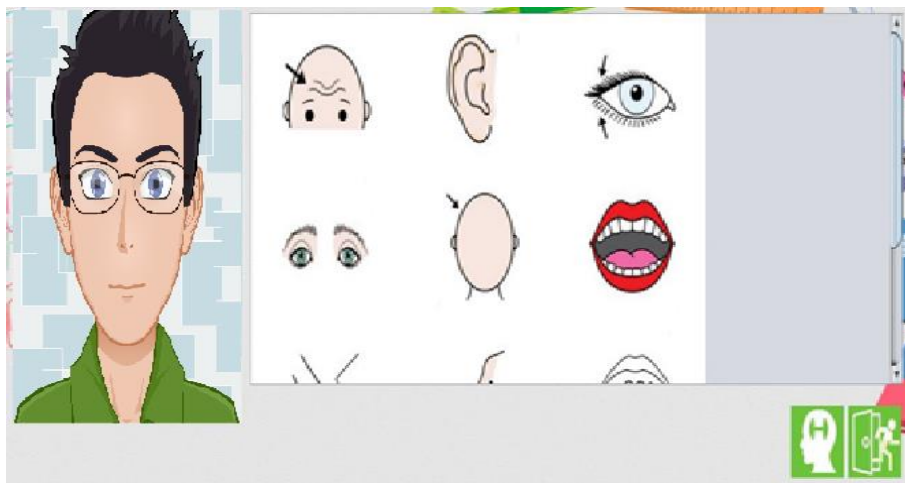


Fig. 8. Reinforcement exercise related to a Nature Exploration course.

At the end of the dynamics of interaction by students and teachers with the virtual tutor, was asked by 5 questions the appreciation of them with respect to the prototype to collect their user experience, the first 3 of yes/no type, the fourth with two possible answers, fun and boring, and for the last, the answer was left open. These questions were the following ones:

1. Do you know how to use a computer (basic level)?
2. Does your elementary school have a computer room?
3. Do you know what a virtual tutor is?
4. What did you think of the prototype of the virtual tutor?
5. How do you think a virtual tutor should be?

In total, 147 students, 7 were 5 years old (5%), 125 were 6 years old (85%) and 15 were seven years old (10%) interacted with the virtual tutor, 72 boys (49%) and 75 girls (51%). Table 1 shows the percentages in the collected answers for the first four mentioned questions. Mainly, for the last question students coincided in their answers saying that a virtual tutor should be friendly, human-looking, colored and not scary. From the answers, it is verified that the user experience was positive and the proposed virtual tutor fulfills his task as a support tool within the teaching-learning process, being welcome and well appreciated by teachers and students.

Table 1. Answers and their percentages for questions to collect the user experience.

Question	Answer 1	Answer 2
1	Yes (40%)	No (60%)
2	Yes (80%)	No (20%)
3	Yes (25%)	No (75%)
4	Funny (94%)	Boring (6%)

5 Conclusions and Future Work

Nowadays, there are many interesting and sophisticating research works about virtual tutors in support of children education, either in the teaching of a new language, as a help in learning for children with disorders such as autism, among others such as technologies for kindergarten children, and tools to teach them to read. Virtual environments for children education both in Mexico and in other countries have been created since the last three decades. However, it is always interesting the way in which such works are addressed, so the application areas can represent an innovation and contribution to the research field [24].

In this paper, a novel and interactive prototype of a virtual tutor was presented, which can be used as a support tool within the teaching-learning process in the first three grades of primary education in Mexico. For this, activities were created in which students come to reinforce the knowledge that is taught within the classroom. In this way, it is possible to bring information and communication technologies closer to children in the form of a human-computer interaction solution.

The presented virtual tutor has as main purpose to support the reinforcement of subjects taught in class, through reinforcement exercises, which allows generating a social and emotional interaction, managing to foster a natural and real-time relationship with students. To the authors' knowledge, MeIPeAS as a virtual tutor is unique and original in the pedagogical agent research field and one of the first created in Mexico for children education purposes.

It is worth mentioning that in its first prototype attempts, the virtual tutor here presented was made known in the 2014 Mexican Space of Science and Technology: The Era of Information and Communication Technologies, during the Workshop of Sociable Robots and Virtual Tutors on October 12, 2014. This event was held in the city of Toluca, State of Mexico in Mexico. From that day, the prototype was perfected and extended until achieving in 2018 what was described and presented here. This paper is one of the first related publications.

Proposing a prototype of an educational virtual tutor as a support tool that contributes and positively impacts educational didactics at the primary level in Mexico, opens the way to reflect and imagine new scenarios of information and communication technologies application [25] in conjunction with the exploitation of serious games and autonomous agents within the educational field [26] [27].

As work in the future, an extension of this prototype can be made to cover the other three missing grades of primary education in Mexico. In addition, other variants are possible to venture into other levels of education as at the university level, where the subjects of study are more complex, but are not exempt from being reinforced with the help of a virtual tutor.

As in similar works, a future extension for the prototype here exposed includes using the virtual tutor to provide support for self-explanation. This extension would be in terms of dialogues with students, where the agent makes questions to guide students in the teaching-learning process. Additionally, other extension is concerned with the use of a mixed short-term and long-term memory mechanism to create dialogues according to the progress of the students and what the virtual tutor remembers to provide customized feedback to them.

References

1. Michael, D.R., Chen, S.L.: Serious games: Games that educate, train, and inform. Muska & Lipman/Premier-Trade (2005)
2. Wouters, P., Van Nimwegen, C., Van Oostendorp, H., Van Der Spek, E.D.: A meta-analysis of the cognitive and motivational effects of serious games. *Journal of educational psychology* 105(2), 249–265 (2013)
3. Huang, R., Spector, J.M., Yang, J.: *Learner Experiences with Educational Technology*. Educational Technology. Springer, Singapore, pp. 91–105 (2019)
4. Passey, D., Dagienė, V., Atieno, L.V., Baumann, W.: Computational Practices, Educational Theories, and Learning Development. *Problemos*, pp. 24–38 (2019)
5. Johnson, W.L., Rickel, J.W., Lester, J.C.: Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education* 11(1), 47–78 (2000)
6. Morton, H., Jack, M.A.: Scenario-based spoken interaction with virtual agents. *Computer Assisted Language Learning* 18(3), 171–191 (2005)
7. Mitrovi, A., Suraweera, P.: Evaluating an animated pedagogical agent. In: *International Conference on Intelligent Tutoring Systems*, Springer, Berlin, Heidelberg, pp. 73–82 (2000)
8. Ashoori, M., Miao, C., Goh, E.S.: Toward a Model of Intelligence in Pedagogical agents (2009)
9. Ki, Y., Baylor, A.L.: Research-based design of pedagogical agent roles: A review, progress, and recommendations. *International Journal of Artificial Intelligence in Education* 26(1), 160–169 (2016)
10. Wise, B., Cole, R., Van Vuuren, S., Schwartz, S., Snyder, L.: Learning to read with a virtual tutor: Foundations to literacy. In: Ngampatipatpong and B. Pellom: *Interactive literacy education: Facilitating literacy environments through technology*, pp. 31–75 (2005)
11. Ward, W., Cole, R., Bolaños, D., Buchenroth-Martin, C., Svirsky, E., Vuuren, S.V., Becker, L.: My science tutor: A conversational multimedia virtual tutor for elementary school science. *ACM Transactions on Speech and Language Processing (TSLP)* 7(4), 1–29 (2011)
12. Atkinson, R.K., Mayer, R.E., Merrill, M.M.: Fostering social agency in multimedia learning: Examining the impact of an animated agent’s voice. *Contemporary Educational Psychology* 30(1), 117–139 (2005)
13. Blair, K., Schwartz, D.L., Biswas, G., Leelawong, K.: Pedagogical agents for learning by teaching: Teachable agents. *Education Technology* 47(1), 56–61 (2007)
14. Song, D.: Designing a Teachable Agent System for Mathematics Learning. *Contemporary Educational Technology* 8(2), 176–190 (2017)
15. Johnson, W.L., Lester, J.C.: Face-to-face interaction with pedagogical agents, twenty years later. *International Journal of Artificial Intelligence in Education* 26(1), 25–36 (2016)
16. Taub, M., Martin, S.A., Azevedo, R., Mudrick, N.V.: The role of pedagogical agents on learning: issues and trends. *Handbook of Research on 3-D Virtual Environments and Hypermedia for Ubiquitous Learning*, IGI Global, pp. 362–386 (2016)
17. Cuéllar, G.R.: La Reforma Integral de la Educación Básica en México (RIEB) en la educación primaria: desafíos para la formación docente. *Revista electrónica interuniversitaria de formación del profesorado* 15(1), 51–60 (2012)
18. Guichard, S.: *The Education Challenge in Mexico: Delivering Good Quality Education to All*. OECD Economics Department Working Papers, No. 447, OECD Publishing, Paris (2005)
19. Treviño Ronzón, E., Cruz Vadillo, R.: La Reforma Integral de la Educación Básica en el discurso docente: Análisis desde el ángulo de la significación. *Perfiles educativos* 36(144), 50–68 (2014)
20. Pelgrum, W.J.: Obstacles to the integration of ICT in education: results from a worldwide educational assessment. *Computers & Education* 37(2), 163–178 (2001)

21. Scott, D., Posner, C., Martin, C., Guzman, E.: *The Education System in Mexico*. UCL Press (2018)
22. Boudourides, M.: Constructivism, education, science, and technology. *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie* 29(3), (2003)
23. Abbot, C.: *ICT: Changing education*. Routledge (2003)
24. Sottolare, R.A., Graesser, A., Hu, X., Holden, H. (Eds.): *Design recommendations for intelligent tutoring systems, Volume 1-learner modeling*. US Army Research Laboratory (2013)
25. Mellati, M., Khademi, M.: Technology-based education: Challenges of blended educational technology. *Advanced online education and training technologies*, IGI Global, pp. 48–62 (2019)
26. Martha, A.S., Santoso, H.B.: The Design and Impact of the Pedagogical Agent: A Systematic Literature Review. *Journal of Educators Online* 16(1) (2019)
27. Payr, S.: The virtual university's faculty: An overview of educational agents. *Applied Artificial Intelligence* 17(1), 1–19 (2003)

Modelo difuso para la evaluación de condiciones apropiadas para turismo en litorales de México

César A. Hoil-Rosas, José J. Carbajal-Hernández, Luis P. Sánchez-Fernández,
Alberto D. Dávila-Lamas

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México
cesarhoil_92@hotmail.com, {jcarbajalh, lsanchez}@cic.ipn.mx,
addavlam@gmail.com

Resumen. En la actualidad, el turismo representa una de las actividades económicas de mayor importancia para México, debido a los ingresos anuales que este produce. Sin embargo, la falta de tecnología aplicada en este campo evita el crecimiento del turismo debido a la falta de información de las zonas costeras. En este trabajo, se desarrolla un modelo computacional basado en un Proceso de Jerarquía Analítica por sus siglas en inglés (AHP), con la finalidad de obtener un índice de evaluación de costas. Para este modelo se utilizan parámetros como la temperatura, oleaje, radiación solar, humedad y velocidad del viento, los cuales fueron medidos y analizados. Este índice ayuda a proporcionar información relevante para los turistas que se disponen a viajar a México.

Palabras clave: AHP, turismo, costas, procesamiento de señales, evaluación.

Diffuse Model for the Evaluation of Appropriate Conditions for Tourism in the Mexican Coastlines

Abstract. Currently, tourism represents one of the most important economic activities for Mexico, due to the annual income it produces. However, the lack of technology applied in this field prevents the growth of tourism due to the lack of information on coastal areas. In this work, a computational model based on an Analytical Hierarchy Process (AHP) is developed, with the purpose of obtaining a cost evaluation index. Parameters such as temperature, waves, solar radiation, humidity and wind speed are used for this model, which were measured and analyzed. This index helps to provide relevant information for tourists who are preparing to travel to Mexico.

Keywords: AHP, tourism, coasts, signal processing, evaluation.

1. Introducción

El turismo es una actividad en expansión en el mundo con un crecimiento anual en el número de viajeros y recursos económicos que produce esta actividad [1]. En los

últimos años el turismo ha adquirido una importancia relevante como generador de ingresos y como fuente de empleo, en especial para las economías emergentes a nivel mundial. Según la OMT [2], las llegadas internacionales anuales superan los 800 millones en todo el mundo, generando ingresos de aproximadamente 800 billones de dólares por año.

En este aspecto, el turismo en México es una de las actividades económicas más relevantes del país y es una de las más importantes en el mundo, ocupando el sexto lugar a nivel mundial en llegadas de turistas internacionales con 42 millones de visitantes en 2018 [3], lo cual contribuyó con el 8.2 % del PIB nacional; cabe mencionar que el 45% de esta actividad está orientada a la zona costera [4].

A lo largo de la historia el turismo ha impactado sobre todo lo que ha tenido contacto, hoy en día las ciudades enfrentan retos complejos al lidiar con ella como el aumento de la población mundial y la transformación digital. Ante esta situación las ciudades y sus gobernantes deben diseñar nuevas estrategias para mejorar aspectos como la planificación y gestión de sus territorios, con la finalidad de garantizar un desarrollo sostenible, al minimizar efectos negativos en el medioambiente, fomentar el bienestar económico y elevar la calidad de vida de los pobladores.

En la literatura se pueden encontrar proyectos en los cuales se realizan clasificaciones de zonas turísticas de acuerdo con determinadas variables como son la urbanización, el riesgo, entre otras. Ranieri en 2016 [5] analiza las condiciones oceanográficas en las playas de Salinópolis y actividades relacionadas con el ocio y la ocupación humana, con el objetivo de categorizar en términos de urbanización. En otro estudio, Irene en 2010 [6] presenta un artículo en el cual se realiza la evaluación subjetiva de los riesgos ambientales asociados con un puerto turístico al sur de Italia, comparando los resultados para proporcionar estimaciones de seguridad ambiental. Estos trabajos analizan y clasifican de acuerdo con datos obtenidos de manera subjetiva por lo cual los resultados se ven afectados por la percepción de cada autor.

Por otro lado, diversos investigadores se han enfocado en la evaluación del paisaje en las zonas costeras. En Brasil, Cristiano (2016) [7], evaluó el paisaje costero del municipio de Torres, tomando en cuenta 26 parámetros naturales y humanos para clasificar las playas en 5 clases, siendo la clase 1 la de mayor atractivo paisajístico y la clase 5 la menos atractiva. Los resultados pusieron de manifiesto una mala gestión de las zonas costeras, con gran cantidad de basura en las playas, instalaciones de apoyo turístico situadas en lugares inadecuados y vertidos de aguas residuales. En 2004, Ergin [8] utilizó 26 parámetros para evaluar los paisajes de las costas en Turquía y Reino Unido, a través de encuestas a turistas y expertos en percepción del paisaje. Después para el año 2010 [9], realizó la evaluación de 34 zonas costeras en la región occidental del Mar Negro.

Los trabajos anteriores presentan un gran avance en el tema de las evaluaciones de zonas costeras. Sin embargo, ninguno está enfocado en crear una herramienta que tome en consideración las prioridades que tiene cada visitante para ayudar al crecimiento del turismo. Además, la falta de tecnología empleada para realizar las mediciones de los diferentes parámetros involucrados evita realizar evaluaciones objetivas.

Con el propósito de establecer las condiciones apropiadas para el turismo en los litorales de México, el presente proyecto propone la creación de un índice de evaluación de costas utilizando un modelo de prioridades basado en un Proceso de Jerarquía Analítica (AHP) y el procesamiento de parámetros climáticos y oceanológicos. Este

índice tiene como principal ventaja el adaptarse a las necesidades de los diferentes turistas que visitan una misma zona ya que permitirá proporcionar información necesaria para los visitantes que se disponen de viajar a las playas de México, permitiendo encontrar los periodos adecuados y los sitios que sean de su mayor interés.

2. Requerimientos

Hoy en día existen diversos criterios para establecer las actividades que se pueden realizar en una zona determinada, los cuales se basan en las condiciones ideales de un conjunto de parámetros. Debido a esto, la siguiente sección explica las condiciones necesarias para llevar a cabo determinadas actividades en las zonas costeras.

2.1. Análisis de factores

Normalmente, los factores de análisis son de gran relevancia para solucionar un problema y estos a su vez constan de varios parámetros que definen su importancia. En este proyecto se consideran 4 factores principales. El primero es la recreación familiar, debido a que ciertas playas no cuentan con las condiciones necesarias para ser visitadas por familias. El segundo son las playas vírgenes, este factor se refiere a la cantidad de infraestructura disponible en el lugar y que facilita las necesidades básicas de los turistas (hotelería, sanitarios, comunicaciones, etc.). El siguiente es la seguridad, el cual es fundamental para brindar protección a turistas que desconocen la zona. Finalmente, la realización de actividades acuáticas, el cual proporciona un nivel de diversión deseada al practicar actividades como el surf, buceo, pesca, etc. De esta forma los factores lo podemos identificar de la siguiente manera:

- F1: Recreación Familiar,
- F2: Playas vírgenes,
- F3: Seguridad,
- F4: Actividades acuáticas.

2.2. Análisis de parámetros

El primer análisis está basado en los parámetros que integran cada factor. Como se mencionó con anterioridad, estos tienen diferente importancia dependiendo de las necesidades del turista ya que no es la misma percepción de aquellos que viajan en familia a lo que prefieren sitios escondidos en lugares vírgenes. En la Tabla 1, se muestran una clasificación de algunos parámetros con base en aquellos que pueden ser medidos con sensores y otros que solo pueden ser evaluados mediante encuestas.

En este trabajo, el análisis de los parámetros que pueden ser automatizados es un conjunto de estudio, ya que puede obtenerse información sobre ellos y establecer las reglas sobre los niveles máximos o mínimos que deben de presentar para obtener un nivel tanto de seguridad como de aceptación del turista. En la Tabla 2, se presenta la clasificación de los parámetros con base en su rango óptimo, así como las tolerancias para su medición. Estos valores fueron obtenidos de otros trabajos de investigación especializados en el turismo [10-12].

Tabla 1. Tipos de parámetros involucrados en el estudio de preferencias.

Institucional	Automatizado	Encuesta
- Crimen	- Temperatura	- Color del agua
- Guardavidas	- Oleaje	- Tipo de arena
- Contaminación	- Radiación solar	- Paisaje
- Venta informal	- Humedad	- Tipo de vegetación
- Mantenimiento	- Velocidad del viento	- Atardecer

Tabla 2. Valores óptimos y límites difusos de cada parámetro.

Parámetro	Rango Óptimo	Unidades	Límites tolerables			
			<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
Temperatura (<i>t</i>)	25 - 30	°C	22	25	30	33
Oleaje (<i>o</i>)	0.5 - 2	<i>m</i>	0	0.5	2	3
Radiación solar (<i>r</i>)	6.1 – 10.1	<i>kW/m²</i>	6.1	8.1	10.1	0
Humedad (<i>h</i>)	40 – 70	%	40	55	70	0
Velocidad del viento (<i>v</i>)	4 -12	<i>Km/hr</i>	1	4	12	16

3. Modelo de evaluación

3.1. Entradas difusas

Para realizar el análisis de cada parámetro, se propone la evaluación previa mediante una función difusa, misma que permita normalizar los niveles de medición de cada parámetro en un rango de [0–1]. Con el fin de evaluar cada parámetro, se han desarrollado las siguientes funciones difusas, tomando en cuenta los límites tolerables (*a, b, c, d*) establecidos en la Tabla 2:

$$\mu(t) = \max \left\{ \left(\frac{x - 22}{25 - 22}, 1, \frac{33 - x}{33 - 30} \right), 0 \right\}, \quad (1)$$

$$\mu(o) = \max \left\{ \left(\frac{x - 0}{0.5 - 0}, 1, \frac{3 - x}{3 - 2} \right), 0 \right\}, \quad (2)$$

$$\mu(r) = \max \left\{ \left(\frac{x - 6.1}{8.1 - 6.1}, 1, \frac{10.1 - x}{10.1 - 8.1} \right), 0 \right\}, \quad (3)$$

$$\mu(h) = \max \left\{ \left(\frac{x - 40}{55 - 40}, 1, \frac{70 - x}{70 - 40} \right), 0 \right\}, \quad (4)$$

$$\mu(v) = \max \left\{ \left(\frac{x - 1}{4 - 1}, 1, \frac{16 - x}{16 - 12} \right), 0 \right\}. \quad (5)$$

3.2. Proceso analítico jerárquico (AHP)

En este proyecto se utiliza el modelo AHP propuesto por Saaty [13], para realizar la evaluación de los litorales turísticos. Esta metodología asigna un nivel de importancia

a cada parámetro dependiendo del factor que se esté evaluando, la interpretación de las escalas se muestra en la Tabla 3.

Tabla 3. Escala propuesta por Saaty en 1980 [14].

			M		F		M		E
	Ig		od		ue		uy		xt
Importancia	ua	+	er	+	rt	+	F	+	re
	l		ad		e		ue		m
			o				rt		o
							e		
Escala	1	2	3	4	5	6	7	8	9

Posteriormente, se crea una matriz pareada ($A = a_{ij}$) donde las prioridades de importancia dadas pueden ser asignadas entre parámetros ($\frac{w_a}{w_b}$). Esta matriz es cuadrada, recíproca y positiva; y se construye como se muestra a continuación:

$$\begin{aligned}
 A &= \begin{bmatrix} P_1 & P_2 & \dots & P_n & P_1 & P_2 \\ \vdots & P_n & [a_{11} & a_{12} & \dots & a_{1n} & a_{21} & a_{22} & \dots & a_{2n}] \\ \vdots & a_{n1} & a_{n2} & \dots & a_{nn} & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\
 &= \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \dots & \frac{w_1}{w_n} & \frac{w_2}{w_1} & \frac{w_2}{w_2} & \dots & \frac{w_2}{w_n} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{6}
 \end{aligned}$$

Una vez elaborada la matriz, se procede a calcular lo que se denomina como eigenvector [15], el cual ayudará a determinar la prioridad de los pesos. Lo cuales serán usados para calcular el índice de evaluación de costas. Con base en lo anterior, la matriz cuadrada **B** se calcula de la siguiente manera:

$$\begin{aligned}
 B &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{21} & a_{22} & \dots & a_{2n} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\
 &\times \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{21} & a_{22} & \dots & a_{2n} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{7}
 \end{aligned}$$

La sumatoria de las filas es determinada de acuerdo con:

$$C_i = \sum_{j=1}^n B_{ij} \forall i = 1, 2, \dots, n. \tag{8}$$

Finalmente, la importancia de los pesos es obtenida normalizando el vector C_i , como se muestra en la siguiente fórmula:

$$W_i = \frac{C_i}{\sum_{j=1}^n C_j} \forall i = 1, 2, \dots, n. \tag{9}$$

Una consideración importante que ofrece el método de AHP es medir el grado de consistencia de la matriz de comparaciones pareadas. Si el grado de consistencia es aceptable, puede continuarse con el proceso de decisión y si es inaceptable, se debe rehacer la matriz. Este grado puede calcularse por medio de la siguiente fórmula:

$$CR = \frac{\lambda_{max} - n}{(n - 1)RI} \tag{10}$$

donde λ_{max} es el máximo eigenvalor de la matriz pareada, n es el tamaño de la matriz, RI es el índice de consistencia aleatoria y CR es el radio de consistencia. El valor de RI depende del número de elementos que se compara, tal y como se muestra en la Tabla 4 [16].

Tabla 4. Índice de consistencia aleatoria.

Tamaño de la matriz	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.52	0.89	1.11	1.25	1.35	1.4	1.45	1.49

El resultado obtenido por CR nos indica que, si este excede de 0.10 se tiene una matriz inconsistente, por el contrario, si este es menor o igual a 0.10, la matriz tiene un grado de consistencia aceptable.

3.3. Índice de evaluación de costas

El Índice de Evaluación de Costas (IEC) es un valor difuso en un rango de [0-1], el cual se ve afectado de manera negativa por la importancia del peso de cada parámetro, generando de esta forma un puntaje que nos indica que tan buena o mala es la evaluación de la playa para realizar alguna actividad turística. El índice puede describirse de forma matemática por medio de la siguiente fórmula:

$$IEC = \sum_{i=1}^n \mu_i(k)W_i^k, \tag{11}$$

donde n es la cantidad de parámetros, $\mu_i(k)$ es el valor difuso obtenido de las ecuaciones definidas en la sección 3.1; k refiere al parámetro y W es la importancia del peso calculado por el método de AHP. Reemplazando cada parámetro en la ecuación 11, se obtiene la siguiente expresión:

$$IEC = \mu(t)W^t + \mu(o)W^o + \mu(r)W^r + \mu(h)W^h + \mu(v)W^v, \tag{12}$$

donde t se refiere a la temperatura, o al oleaje, r a la radiación solar, h a la humedad y v a la velocidad del viento. Finalmente, el resultado obtenido será el índice de evaluación de costas, donde mientras más cercano esté a “1” será más recomendable el lugar y por el contrario mientras más próximo esté a “0” menos recomendable será el lugar.

4. Resultados y discusiones

Para comprobar la eficiencia del índice de evaluación de costas, se realizó una prueba ejemplificando un viaje familiar, asignando la importancia a los parámetros y factores, que resultan de mayor relevancia en este tipo de viajes. En este caso, las prioridades de los factores elegidos de mayor a menor importancia son la seguridad, recreación familiar, actividades acuáticas y playas vírgenes; mientras que los parámetros más

relevantes para ellos son el oleaje, la radiación solar, la velocidad del viento, la temperatura y la humedad. El procedimiento y los resultados se muestran a continuación.

4.1. Análisis de comparaciones

De acuerdo con el orden establecido por la familia en la sección anterior se establecen las matrices de comparaciones. La Tabla 5 muestra las comparaciones entre factores, definidos con base en los criterios de una familia. También, se muestra una comparación entre el método de normalización y el AHP (Eigenvector) siendo este último más preciso. Las tablas 6, 7, 8 y 9 tienen las respectivas matrices pareadas para cada parámetro, donde las comparaciones se realizan teniendo en cuenta el factor como criterio de decisión.

Tabla 5. Comparación de los Factores con un CR = 0.0348.

Factores	F1	F2	F3	F4	Normalizado	Eigenvector
F1: <i>Recreación familiar</i>	1	5	1/3	4	0.2990	0.2669
F2: <i>Playas vírgenes</i>	1/5	1	1/9	1/2	0.0524	0.0533
F3: <i>Seguridad</i>	3	9	1	6	0.5497	0.5911
F4: <i>Actividades acuáticas</i>	1/4	2	1/6	1	0.0989	0.0887
	Suma:				1	1

Tabla 6. Comparación de los parámetros de acuerdo con F1 y un CR=0.085.

Parámetros	<i>t</i>	<i>o</i>	<i>r</i>	<i>h</i>	<i>v</i>	Normalizado	Eigenvector
(<i>t</i>) Temperatura	1	4	1/3	6	7	0.314	0.286
(<i>o</i>) Oleaje	1/4	1	1/5	3	5	0.1619	0.1197
(<i>r</i>) Radiación	3	5	1	7	9	0.4282	0.5055
(<i>h</i>) Humedad	1/6	1/3	1/7	1	2	0.0624	0.054
(<i>v</i>) Viento	1/7	1/5	1/9	1/2	1	0.0335	0.0348
	Suma:					1	1

Tabla 7. Comparación de los parámetros de acuerdo con F2 y un CR=0.0665.

Parámetros	<i>t</i>	<i>o</i>	<i>r</i>	<i>h</i>	<i>v</i>	Normalizado	Eigenvector
(<i>t</i>) Temperatura	1	3	1/5	1/3	2	0.126	0.1113
(<i>o</i>) Oleaje	1/3	1	1/9	1/6	2	0.0696	0.0578
(<i>r</i>) Radiación	5	9	1	3	7	0.482	0.5245
(<i>h</i>) Humedad	3	6	1/3	1	4	0.2763	0.2535
(<i>v</i>) Viento	1/2	1/2	1/7	1/4	1	0.0461	0.0529
	Suma:					1	1

Tabla 8. Comparación de los parámetros de acuerdo con F3 y un $CR=0.0841$.

Parámetros	<i>t</i>	<i>o</i>	<i>r</i>	<i>h</i>	<i>v</i>	Normalizado	Eigenvector
(<i>t</i>) Temperatura	1	1/7	1/5	3	1/3	0.0816	0.0634
(<i>o</i>) Oleaje	7	1	3	9	5	0.4361	0.5128
(<i>r</i>) Radiación	5	1/3	1	7	3	0.2849	0.2615
(<i>h</i>) Humedad	1/3	1/9	1/7	1	1/5	0.0312	0.0333
(<i>v</i>) Viento	3	1/5	1/3	5	1	0.1662	0.1290
Suma:						1	1

Tabla 9. Comparación de los parámetros de acuerdo con F4 y un $CR=0.05975$.

Parámetros	<i>t</i>	<i>o</i>	<i>r</i>	<i>h</i>	<i>v</i>	Normalizado	Eigenvector
(<i>t</i>) Temperatura	1	1/5	4	4	1/3	0.1705	0.1310
(<i>o</i>) Oleaje	5	1	8	8	3	0.4468	0.5167
(<i>r</i>) Radiación	1/4	1/8	1	1	1/6	0.0454	0.0445
(<i>h</i>) Humedad	1/4	1/8	1	1	1/6	0.0454	0.0445
(<i>v</i>) Viento	3	1/3	6	6	1	0.2919	0.2633
Suma:						1	1

Para obtener los pesos finales de cada parámetro es necesario multiplicar los pesos obtenidos en la Tabla 5 por los pesos individuales de cada parámetro con respecto a cada factor. La Tabla 10 muestra los valores finales obtenidos por el método de AHP.

Tabla 10. Comparación de los parámetros considerando todos los factores y pesos.

Parámetro Factor						Pesos de factores					
	<i>t</i>	<i>o</i>	<i>r</i>	<i>h</i>	<i>v</i>		<i>t</i>	<i>o</i>	<i>r</i>	<i>h</i>	<i>v</i>
F1	0.2860	0.1197	0.5055	0.054	0.0348	0.2669	0.0763	0.0319	0.1349	0.0144	0.0093
F2	0.1113	0.0578	0.5245	0.2535	0.0529	0.0533	0.0059	0.0031	0.0280	0.0135	0.0028
F3	0.0634	0.5128	0.2615	0.0333	0.1290	0.5911	0.0375	0.3031	0.1546	0.0197	0.0763
F4	0.1310	0.5167	0.0445	0.0445	0.2633	0.0887	0.0116	0.0458	0.0039	0.0039	0.0234
Score Final						1	0.1314	0.3840	0.3214	0.0516	0.1117

4.2. Medición de parámetros

Las mediciones de los parámetros fueron realizadas durante los días del 2 de junio al 12 de junio del 2019 en la ciudad de La Paz, Baja California Sur. La base de datos se formó al medir los parámetros cada 15 minutos durante 11 días, obteniendo 96 muestras diarias y un total de 1056 muestras por cada uno. Los parámetros medidos son la temperatura, el oleaje, la radiación solar, la humedad y la velocidad del viento. La Figura 1, muestra gráficamente el comportamiento de cada uno de estos, así como los límites permitidos que se establecieron en la Tabla 2.

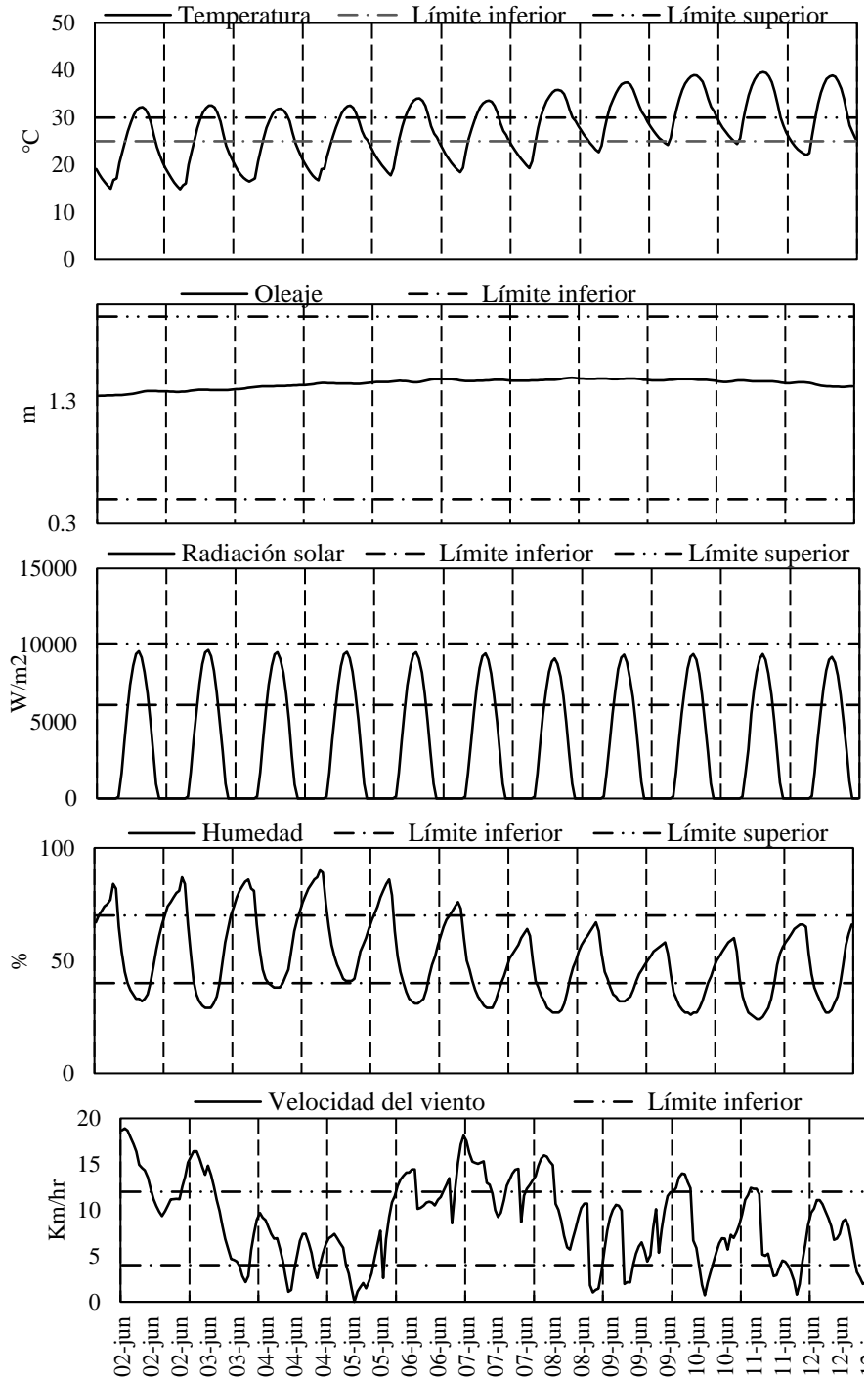


Fig. 1. Gráficas de cada parámetro durante los días del 2 al 12 de junio de 2019.

En la Figura 1, se pueden apreciar un comportamiento aleatorio en cada parámetro, los cuales sobrepasan los límites óptimos establecidos para cada uno; a excepción del oleaje el cual siempre se mantiene dentro del rango óptimo.

4.3. Índice de medición de costas

Finalmente, para obtener el Índice de Medición de Costas se utiliza la ecuación 12, tomando en cuenta los pesos obtenidos en la Tabla 10, así como el resultado de la evaluación de cada parámetro de acuerdo a las funciones difusas establecidas en las ecuaciones 1-5.

En la Figura 2 se muestra el resultado de la evaluación de la costa de la Paz, Baja California Sur. Como se observa, tuvo una evaluación regular debido a que la mayoría de los parámetros estaban fuera del rango permitido. Sin embargo, como se muestra en la Tabla 10 el oleaje es el que tiene mayor peso y esto hace que el índice sea regular ya que este parámetro siempre se mantuvo en el rango permitido. Analizando la gráfica por día, observamos que la mejor evaluación se da en los días del 9 al 12 de junio de 2019, esto se debe a que la humedad y la velocidad del viento, permanecieron en mayor medida en su rango óptimo; por lo cual el factor principal establecido en la sección 4 por la familia, que es la seguridad, es mejor evaluada, obteniendo así una mayor calificación en el índice.

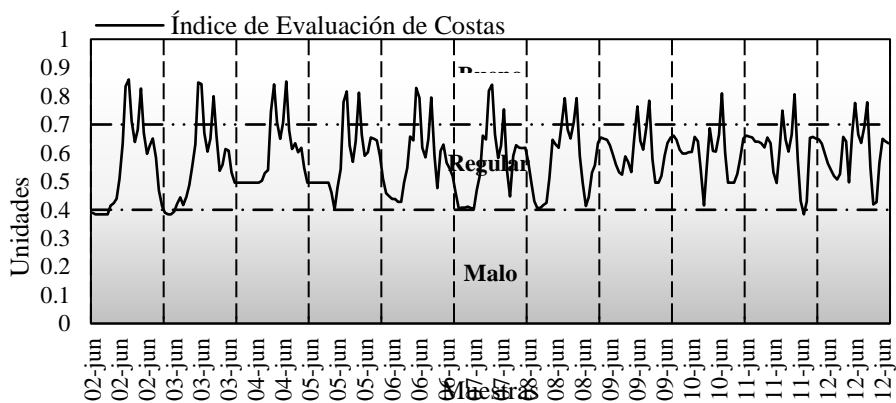


Fig. 2. Gráfica del índice de evaluación de costas para La Paz, Baja California Sur.

5. Conclusión y trabajo a futuro

El índice de evaluación de costas presente en este trabajo es una herramienta novedosa ya que es adaptativa a las necesidades de cada turista y a los parámetros presentes en cada costa. Este índice ayuda a los turistas a tener una mayor información de la playa a visitar, que les ayudará a encontrar el periodo y la costa que mejor se adapte a sus necesidades.

De acuerdo con la literatura, existen trabajos similares que se encargan de evaluar diferentes características de una playa; sin embargo, nos están enfocados en ayudar al turismo, por lo cual este modelo es innovador ya que conjunta todos los factores que el turista necesita para indicarle si las condiciones en playa son buena o mala de acuerdo

con sus preferencias. Como trabajo a futuro, se plantea el aumentar el número de parámetros a medir, integrando otras técnicas y modelos computacionales para obtener un mejor análisis y evaluación de las costas.

Referencias

1. Macário de Oliveira, V., Pasa Gómez, C., Ataíde Cândido, G.: Indicadores de sustentabilidad para la actividad turística. Una propuesta de monitoreo usando criterios de análisis. *Estudios y perspectivas en turismo* 22(2) (2013)
2. OMT-Organización Mundial del Turismo: *Tourism highlights 2006 Edition*. Madrid (2006)
3. SECTUR: *Visión Global del Turismo a México. Análisis de mercados, perspectivas del turismo mundial* (2018)
4. SECTUR: *Turismo de internación 2001-2005, Visitantes internacionales hacia México* (2006)
5. Ranieri, L.A., El-Robrini, M.: *Condição Oceanográfica, Uso e Ocupação da Costa de Salinópolis (Setor Corvina-Atalaia), Nordeste do Pará, Brasil*. *Revista de Gestão Costeira Integrada* 16(2), 133–146 (2016)
6. Irene, P., Paolo, V., Donatella, V., Alberto, M.J., Mauro, F., Giovanni, Z.: Mapping the environmental risk of a tourist harbor in order to foster environmental security: objective vs. subjective assessments. *Marine pollution bulletin* 60(7), 1051–1058 (2010)
7. Cristiano, S.D.C., Rockett, G.C., Portz, L.C., Anfuso, G., Gruber, N.L.S., Williams, A.T.: Evaluation of Coastal Scenery in Urban Beach: Torres, Rio Grande do Sul, Brazil. *Revista Gestão Costeira Integrada Journal of Integrated Coastal Zone Management*. Lisboa: Associação Portuguesa dos Recursos Hídricos, Vol. 16, n. 1, pp. 71–78 (2016)
8. Ergin, A., Karaesmen, E., Micallef, A., Williams, A.T.: A new methodology for evaluating coastal scenery: fuzzy logic systems. *Area*, 36(4), 367–386 (2004)
9. Ergin, A., Özölçer, İ.H., Şahin, F.: Evaluating coastal scenery using fuzzy logic: Application at selected sites in Western Black Sea coastal region of Turkey. *Ocean Engineering* 37(7), 583–591 (2010)
10. Kim, D.I., Matsuyama, Y., Nagasoe, S., Yamaguchi, M., Yoon, Y.H., Oshima, Y., Honjo, T.: Effects of temperature, salinity and irradiance on the growth of the harmful red tide dinoflagellate *Cochlodinium polykrikoides* Margalef (Dinophyceae). *Journal of Plankton Research* 26(1), 61–66 (2004)
11. Gilmore, J.W.: Cálculo y mapeo de la radiación solar directa y difusa en Costa Rica. *Uniciencia* 22(1), 55–69 (2008)
12. Cengel, Y.A., Boles, M.A., Buesa, I.A.: *Termodinámica*, vol. 10. São Paulo: McGraw-Hill (2006)
13. Saaty, T.L.: Decision making with the analytic hierarchy process. *International journal of services sciences* 1(1), 83–98 (2008)
14. Saaty, T.L., Wind, Y.: Marketing applications of the analytic hierarchy process. *Management science* 26(7), 641–658 (1980)
15. Perron, O.: Grundlagen für eine Theorie des Jacobischen Kettenbruchalgorithmus. *Math. Ann.* 64, 11–76 (1907)
16. Toskano, G., Bruno, G.: El proceso de análisis jerárquico (AHP) como herramienta para la toma de decisiones en la selección de proveedores. Trabajo de grado, Universidad Nacional Mayor de San Marcos, Lima, Perú (2005)

Multilevel Image Transmission using Polar Codes

Guadalupe Donaji Aguilar Licea, Saul Lazcano Salas

Universidad Autónoma del Estado de México, CU Valle de México, Mexico
gpe.ag.18@gmail.com, slazcanos@uaemex.mx

Abstract. Nowadays, with the big amounts of data that gets transmitted all the time through the communication systems, and among the different technologies that are in development in order to support future networks, many methods of information processing such as channel coding strategies and multilevel image transmission are being studied in order to be used with the oncoming technologies. One of the most common problem in communication systems, is how intrusive noise inherent to the channel during the transmission can be, as well as the limited bandwidth available, especially in wireless channels. This work presents a multilevel image decomposition using a wavelet transform, and their transmission through an AWGN channel with polar codes as a channel coding strategy, using different frame sizes ($N=1024$, $N=512$ and $N=256$) in order to test how robust polar codes' performance can be for multilevel image decomposition in presence of noise during transmission.

Keywords: wavelet, multilevel image decomposition, polar codes, channel coding.

1 Introduction

The need to improve communication systems for the exchange of information, has led to different techniques to be created. In terms of 5G networks, these techniques become paramount, since most of the services expected from it rely on high efficiency and reliability [1]. One of these improvements comes from the work of C. Shannon in 1948, which marks a starting point for the emergence of a new area of work: channel coding, which involves techniques that allow the detection and / or correction of errors that may exist during the transmission process. This can be done in different ways, as long as the encoder and therefore the decoder, are able to identify the errors and correct them afterwards.

Following this line, in 2009, Erdal Arıkan made a proposal of a channel coding technique [2], which involves the polarization of the channel, hence calling it Polar Coding. Channel polarization takes the original channel and combines it in order to separate the good channels (noiseless) from bad channels (with high noise) and then, use good channels to transmit the information. This process allows Polar Codes to reach a BER performance close to Shannon's limit using a low complexity coding scheme, making them so attractive to use as a standard channel coding technique for wireless communications, such as 5G and others.

This paper focuses on analyzing the strengths of Polar Codes in terms of multi-level transmission of wavelet processed images. Wavelets, in image processing, are

algorithms that make a decomposition of the image and divides it in four different components, each one containing different details. For this analysis, two wavelets were used in order to make a comparison; these wavelets were: Haar [3] because it is the simplest and most common in image processing, making it, in most platforms, the default. The other one was Daubechies 15 [4], due to this being an extension of the Haar wavelet.

The remaining contents of this paper are organized as follows. Section 2 gives some background on wavelets, and a description of polar codes. Section 3 describes the testing scheme for the polar codes. The results of the simulation are shown and described in Section 4. And finally, Section 5 presents the conclusions of the work along with possible future implementation of this study.

2 Wavelets and Polar Codes

Wavelets, for being mathematical functions, have been developed in different areas, in image processing they allow to process the information into different frequency components and analyze any discontinuity within the image. Some of these wavelets are derivatives of the Haar and Daubechies transforms. In general, these wavelets decompose the signals in half, having as a result two smaller signals, making it a very simple, fast, and easier to revert.

EZW (Embedded Zerotrees Wavelet) is another algorithm used in image compression [5]. It classifies the transformed coefficients in a tree, where the low frequency coefficients are placed at the root and the higher frequencies get placed at the next subbands of the tree. In [6] an EZW-Polar Code image coding method is proposed in order to help finding more applications for image coding and polar codes.

In other way, as mentioned previously, Polar Codes separate the noisy channels from the noiseless ones, a key step to construct Polar Codes is channel polarization, which is linked to a specific channel parameter (error probability - P_e), in other words, if the P_e changes, polarization changes too, Polar Codes are not universal. In this work, we build Polar Codes according to SNR point selected.

3 Test Scheme

The flow chart in Fig. 1 shows the process of the testing system used for both wavelet cases, which includes the transmission system, and the treatment of the image used (Lena 512 X 512). It should be mentioned that the frame lengths used are 1024 bits, 512 bits, and 256 bits, making the channels' lengths 2048, 1024 and 512 respectively.

As it is shown, in order to transmit the image using polar codes, first, it had to be decomposed using the wavelet, which in this case, consisted of a one-level decomposition, see Fig. 2. Afterwards, each resulting image was converted into a vector, so it could be worked with during the simulations.

For all transmissions, an AWGN (Additive White Gaussian Noise) channel was chosen, along with a SNR (Signal to Noise Ratio) that would be just the same as the channel state; in this case, it ranged from 1 to 10 with $\Delta = 0.1$. This with the purpose of

having optimal channel states and SNR at all times and ensure as most as possible, the integrity of the transmitted information. Because the channel is already established, as well as its parameters, it means that the polar codes used for this scheme, are not universal [5].

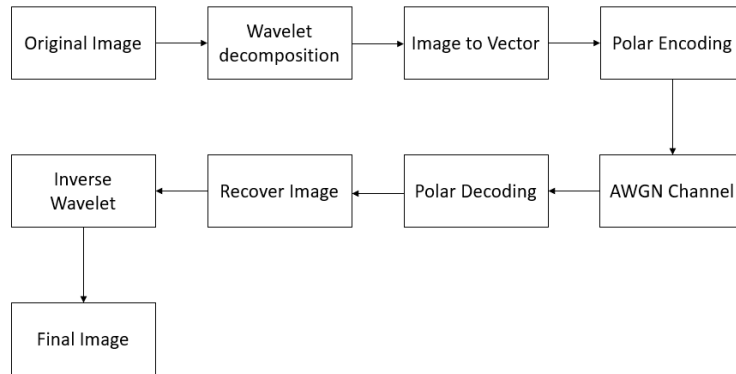


Fig. 1. Transmission system.

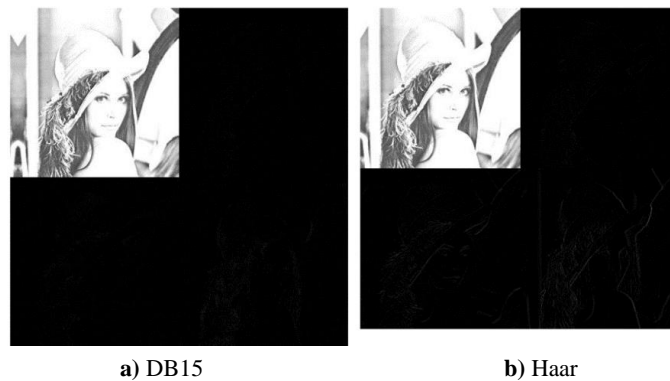


Fig. 2. Images after wavelet composition.

4 Results

A comparison of the BER (Bit-Error Rate) and FER (Frame-Error Rate) obtained from the transmission simulation of both wavelets and three frame lengths is illustrated in Fig. 3 to Fig. 5. For the 1024 bits frame, in both wavelet cases, once the SNR reached +3dB, the error correction became almost optimal, correcting all errors, with the exception of the DB15 wavelet, where it found 8 BERs and 1 FERs when the SNR channel=3.8dB, from then on, the rest of the channel states did not encounter more errors.

A similar situation occurred for N=512, where the error correction became optimal at +3.4dB but for the Haar wavelet, at the SNR channel=3.7dB the BER = 8 and FER

=1 and again, from then on, the channels became optimal. For $N=256$, the error correction became optimal until the SNR reached +4dB.

Once the transmission simulation was over, the resulting images were recovered by reconstructing the vector into the corresponding image matrix.

The generated errors according to the SNR in the previous graphic, can be appreciated in the images as well. Fig. 6 shows an example of the differences in the errors, where the image sent in the 1024 bits frame has almost half of the errors compared to the resulting ones in both the 512 and 256 bits frame, however none of them have errors once the SNR got higher than 3dB and 4dB respectively.

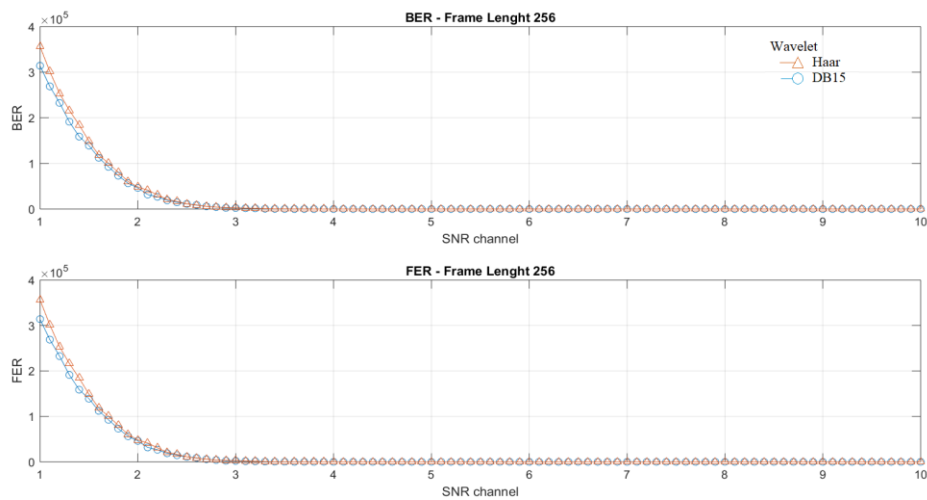


Fig. 3. BER/FER – $N=256$ results.

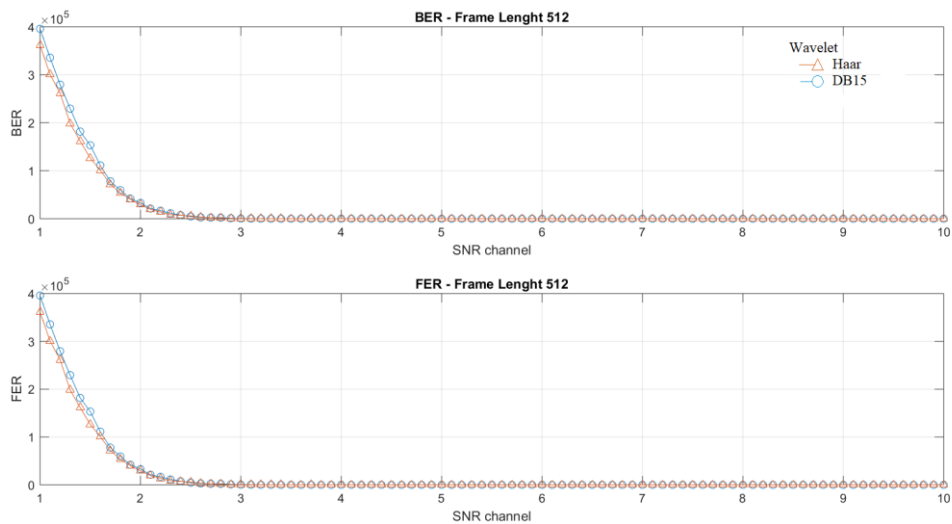


Fig. 4. BER/FER – $N=512$ results.

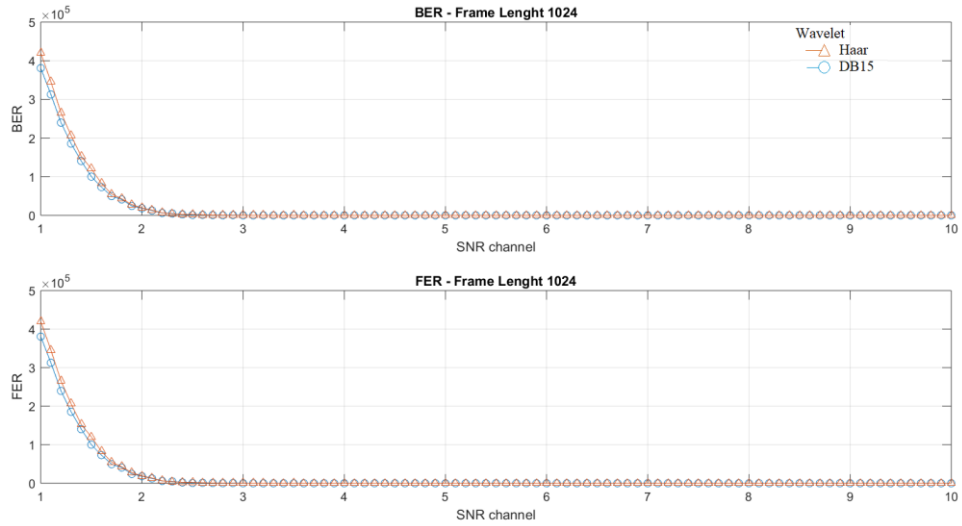


Fig. 5. BER/FER – N=1024 results.

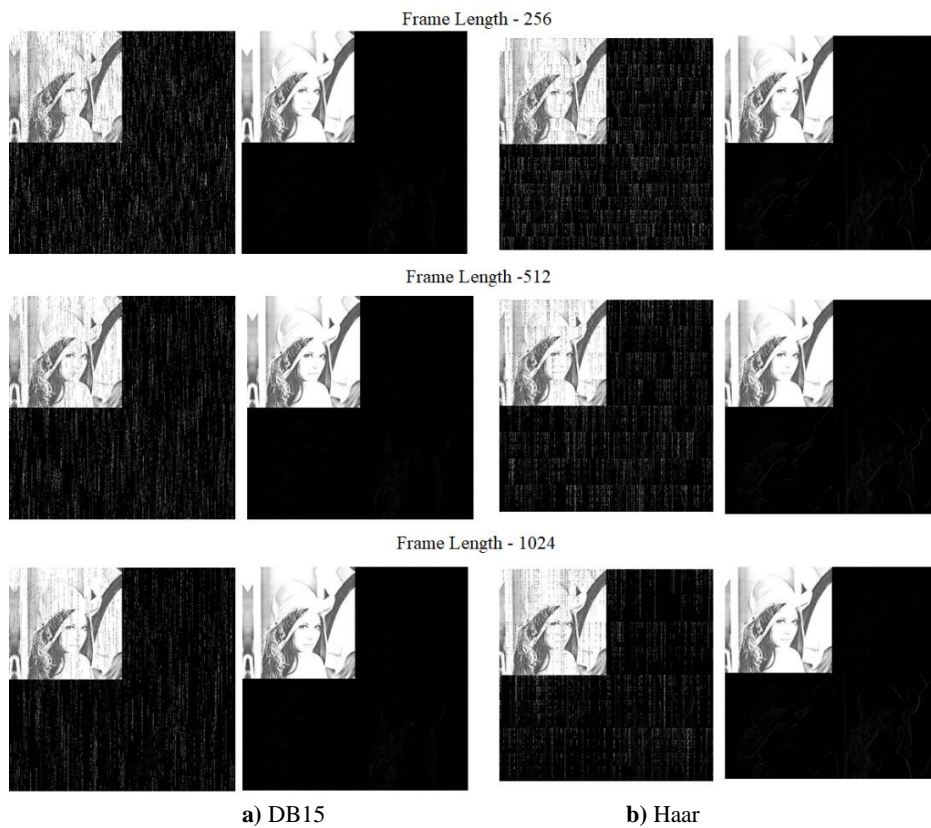


Fig. 6. Images after transmission with and without errors (SNR: 1.7dB and 9.2dB).

To make a comparison between the original image and the transmitted ones, it was necessary to make a reconstruction of these obtained images by applying the corresponding inverse wavelet transform to each one. Then three different comparison parameters, such as, MSER (Mean Square Error Rate), PSNR (Peak Signal to Noise Ratio) and SNR (Signal to Noise Ratio) were used in every image along with the original. These results are displayed in Fig. 7 to Fig. 9.

Once the BER and FER got to zero, the resulting images were identical to each other for each wavelet, as well as the original image. For example, the resulted DB15 image from SNR channel at 3.7dB, is identical to the one obtained from the 9.5dB channel, no matter the frame length. The same situation happened with the Haar wavelet obtained images.

Making an analysis of the results in the previous graphics, it became predictable, in all three cases: $N=256$, $N=512$ and $N=1024$, that once the BER and FER become zero, the MSER, PSNR and SNR would start giving the same results. For $N=256$, with the DB15 wavelet, this started in the SNR channel=4.0dB; for the Haar case, it started in the SNR channel = 4.3dB. When $N=512$, this happened at the SNR channel = 3.5dB except for the Haar wavelet, where in channel 3.7dB, the BER=8 and FER=1.

Finally, for $N=1024$, this situation was presented earlier, for Haar it started in the 3.1dB channel, and for DB15 in 3.3db channel, with the exception on channel 3.5, where the FER=1 and BER=8.

These results exhibit a particular behavior of polar codes in terms of performance versus frame lengths, where the smaller the frame the worse the performance, whereas the longer the frame the better the performance gets. [6]

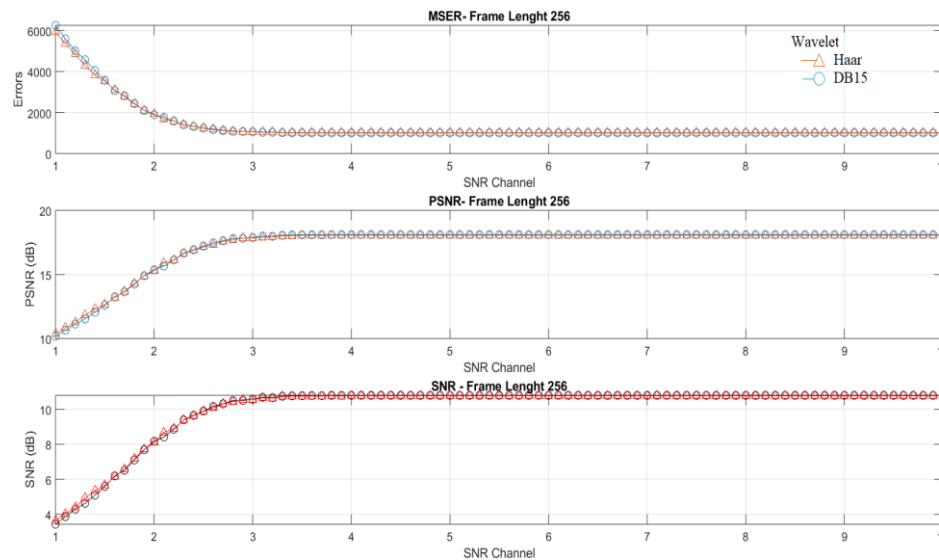


Fig. 7. Image comparison results N=256

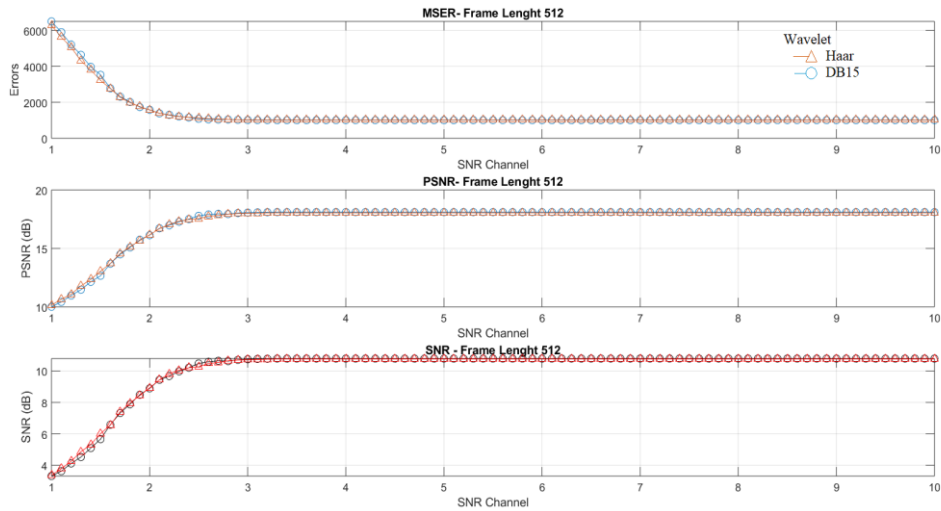


Fig. 8. Image comparison results N=512

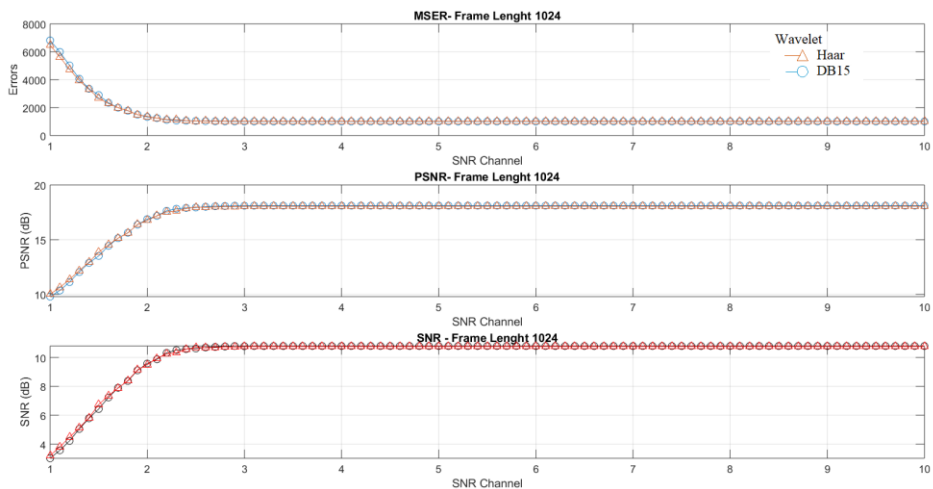


Fig. 9. Image comparison results N=1024

5 Conclusions and Future Work

It can be concluded that since polar codes have better performance with long frames of information opposed to small ones, such as the ones used nowadays (256 bits), this coding technique is a strong option for 5G, due to the fact that more information can be transmitted at once, and the optimal channel SNR can be reached more easily during the transmission, meaning that the integrity of the information can be maintained as if it were sent in smaller frames.

This test only included a single level decomposition, but hopefully it can open the door to future studies that could involve the use of the frozen bits or frozen channels,

where the decomposition of the image via wavelets play a more active role, meaning that the wavelets would involve a greater level of decomposition and therefore, the main component of the image is sent through the optimal channels and the rest of the details are sent through the less optimal channels, making the transmission a bit faster by lowering the redundancy, and ensuring that at least the main component will be received without or at most, with minimal errors.

Following this line of transmissions using polar codes, this technique can be implemented in SDR (Software Defined Radio) devices, as well as platforms, being GNU Radio the main platform because of the extensive utilities it offers, and since it is open source, the user can create as many tools as necessary.

References

1. ITU: Document 5D/TEMP/300 (Rev.1)
2. Arikan, E.: Channel polarization: A method for constructing capacity-achieving codes. In: IEEE International Symposium on Information Theory (2008)
3. Haar, A., Zimmermann, G.: On the Theory of Orthogonal Function Systems. *Fundamental Papers in Wavelet Theory*, pp. 155–188. doi:10.1515/9781400827268.155 (2009)
4. Daubechies, I.: Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics* 41(7), 909–996. doi:10.1002/cpa.3160410705 (1988)
5. Shapiro, J.M.: Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing* 41(12), 3445–3462 (1993)
6. Ma, H.M., Chen, D., Zhang, C.: Embedded zerotrees wavelet image coding using source polar codes. In: 2017 IEEE 17th International Conference on Communication Technology (ICCT). doi:10.1109/icct.2017.8359927 (2017)
7. Hassani, S.H., Urbanke, R.: Universal polar codes. In: 2014 IEEE International Symposium on Information Theory. doi:10.1109/isit.2014.6875073 (2014)
8. Payommai, T., Chamnongthai, K.: Performance of polar code for image transmission. In: 2013 International Symposium on Intelligent Signal Processing and Communication Systems. doi:10.1109/ispacs.2013.6704592 (2013)

Prediction of Cancer using Microarrays Analysis by Machine Learning Algorithms

José Luis Velázquez-Rodríguez¹, Yenny Villuendas-Rey²,
Cornelio Yáñez-Márquez¹

¹ Instituto Politécnico Nacional, Centro de Investigación de Computación, Mexico

² Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo,
Ciudad de México, Mexico

Abstract. The analysis of microarrays that contain information on biomolecules related to different types of cancer is one of the current issues in international scientific research due to the impact it has on public health worldwide. The advances in this scientific research route have been impressive; the different international research groups have applied sophisticated algorithms for machine learning, data mining and related branches with the aim of finding solutions to this problem. The present article contains a study of several the classification algorithms used in the literature, and their application for the prediction of cancer using microarrays analysis. More in detail, we tested six classification models, over microarrays data. The application of the supervised classification algorithms was done over the Weka 3 Software environment, using the Leave One Out validation scheme. In addition, a nonparametric statistical test (the Friedman test) identified the significant differences in the performance of the algorithms, according to the experimental results obtained. The analysis of the hypothesis tests of the experimental results indicates that the Support Vector Machine models outperform others for the prediction of cancer.

Keywords: classification models, microarrays, cancer prediction, computational intelligence.

1 Introduction

The analysis of microarrays that contain information on biomolecules related to different types of cancer has great potential for medical diagnostic tests [1]. A microarray contains the gene expression of an individual, which is known through clinical diagnoses if it suffers, or not, a certain type of cancer. It is possible to train Machine Learning algorithms in order to learn the relationship between the levels of gene expression of a patient and their condition as to whether or not they suffer from certain type of cancer; in other words, it is possible to diagnose whether an individual suffers, or not, cancer, through the intelligent analysis of the microarray that contains the information of the level of expression of their genes [2, 3].

The advances in this scientific research route have been impressive [4]; the different international research groups have applied sophisticated algorithms for machine learning, data mining and related branches with the aim of finding solutions to this problem. They have been applied from effective algorithms such as random forests [5],

through extensive comparative studies of dozens of gene classifiers [6], to the application of next-generation neuronal models [7], guaranteeing with this intense activity the validity and current status of this scientific research topic.

In this paper, we review several machine learning algorithmic solutions proposed in recent years to analyze microarray data (section 2). We study some of the classification algorithms used in the literature, and their application to the diagnosis of cancer using microarray data (section 3). In addition, we used 10 microarray datasets (section 4) to test all the classification models. The different datasets have imbalanced data, mixed categorical and numerical attributes. Then, the numerical experiments performed in the comparison between the different classification algorithms applied for different types of cancer diagnoses were made. Finally, we offer the conclusion and some lines of future work (section 5).

2 Background

We present some results that have been reported by international research groups, derived from applying different algorithms of machine learning, data mining and related branches, with the purpose of finding solutions to the problem represented by microarray analysis. that contain information on biomolecules related to different types of cancer.

Korucuoglu et al. [8] proposed a method based on Bayesian Networks, where biological routes are taken for a Bayesian Network and each path is qualified for a method equivalent to Bayesian-Dirichlet. The method was tested with datasets of cancer cells microarrays.

Tan et. al. [9] claimed that by using a method to classify microarrays data it is not only necessary to obtain good results, but also that the results can be easily interpreted, and therefore he proposed a method based on Decision Trees for the analysis of carcinogenic samples, which generates precise and easy to interpret rules.

The genome expression patterns generally consist of thousands of genes, and it is necessary to extract the most significant genes. Cho et. al. [10] obtained significant genes from representative vectors, and proposed a set of neural networks that were trained with this set of significant genes to classify a dataset divided into three classes: leukemia, colon and B-cell lymphoma.

Guyon et al. [11] constructed an efficient classifier for genetic analysis and drug use, using training examples for cancer treatment and using data from normal patients. They propose the use of Support Vector Machines for the elimination of recursive characteristics.

Hu et. al. [12] observed that some types of cancer are related to each other while there are others that are very differentiable. They use a dataset with information on different types of cancer and a clustering method.

In the work of Ruiz et. al. [13] a heuristic is proposed for the selection of attributes to datasets of genetic expressions. His method is based on the use of statistical significance.

Despite the above proposals, there is a lack of experimental comparisons of several classification algorithms over multiple microarray datasets, for cancer pre-diagnosis. In this paper, we address this gap.

3 Datasets and Algorithms

In this section, a summary of applied classification algorithms and datasets related to microarray are presented for the classification of cancer.

3.1 Datasets Related to Microarray

The used datasets include information about different types of cancer, such as leukemia, colon cancer, adenocarcinoma, brain tumor, lymphoma, and others. The datasets used in this paper were taken from online information provided by different repositories. We used the microarray data related to standard classification datasets. In the following, a description of each of the datasets used is shown.

Leukemia dataset: It is a dataset based on the monitoring of gene expressions by DNA microarrays in humans with Acute Leukemia. Distinguish between two types of conditions, between Acute Myeloid Leukemia (AML) and Acute Lymphoblastic Leukemia (ALL). It contains 38 samples of which 27 are ALL and 11 AML, each sample in turn contains 3051 expressions [14].

Colon dataset: The gene expression levels in colon tissue samples are shown in this dataset. It contains 62 patients of which 40 have tumors and 22 are normal. Each sample contains 2,000 human genes [15].

Adenocarcinoma dataset: The original dataset contained 16063 Affymetrix chip genes, but they were reduced to 9,868 by the author of [16]. The Adenocarcinoma dataset includes information on 76 patients, of which 64 have primary tumors and 12 metastatic tumors. The data of 9,868 genes are shown for each patient [16].

Brain Tumor dataset: This dataset contains 42 profiles of genetic expressions of microarrays, divided into five different types of tumors of the central nervous system, which are, 10 medulloblastoma, 10 malignant gliomas, 10 atypical teratoid/rhabdoid tumor, eight primitive neuroectodermal tumors and four of human cerebellum. Each profile contains 5,597 genes [15].

Breast Cancer 1 dataset: This dataset contains information on 78 people who suffered from breast cancer [17]. It includes 34 patients who developed metastases within the first five years (class 1) and 44 patients who remained disease-free for more than five years (class 2).

Breast Cancer 2 dataset: This dataset complies with the two classes of the Breast Cancer 1 dataset previously described, and a third additional class which includes 18 patients with germline mutations in the BRCA1 gene (class 3). This dataset contains 96 patterns divided into three classes [17].

Lymphoma dataset: This dataset contains the gene expression levels of the three most prevalent adult lymphoid neoplasms: 42 samples of diffuse large B-cell lymphoma (class 1), nine observations of follicular lymphoma (class 2) and 11 cases of lymphocytic leukemia chronic (class 3). The total samples are 62 and each contains 4,026 genetic expressions (attributes) [15].

National Cancer Institute (NCI) dataset: The data come from complementary DNA microarrays. This dataset contains 61 samples that can be divided into eight different tumor types samples: seven of breast cancer, five of central nervous system cancer, seven of colon cancer, six of leukemia, eight of melanoma, nine of lung carcinoma, six of ovarian and nine of kidney tumors. Each sample contains 5,244 genes [18].

Prostate cancer dataset: This dataset contains the samples of 52 prostate tumors and 50 non-tumor samples, giving a total of 102 samples and each sample contains 6,033 genes [15].

SRBCT dataset: This dataset contains gene expression profiles for classifying small round blue cell tumors in childhood (SRBCT) and contains four classes (12 samples of neuroblastoma, 20 of rhabdomyosarcoma, eight of non-Hodgkin lymphoma, and 23 of Ewing tumors family), obtained from microarrays containing 2308 genes [15].

Table 1. Description of the datasets used.

	Datasets	Attributes	Imbalance analysis		Classes
			Instances	IR	
1.	Leukemia	3051	38	2.454	2
2.	Colon	2000	62	1.818	2
3.	Adenocarcinoma	9868	76	5.333	2
4.	Brain tumor	5597	42	2.500	5
5.	Breast cancer 1	4869	77	1.330	2
6.	Breast cancer 2	4869	95	2.444	3
7.	Lymphoma	4026	62	4.666	3
8.	National Cancer Institute (NCI)	5244	61	1.800	8
9.	Prostate cancer	6033	102	1.040	2
10.	SRBCT	2308	63	2.875	4

In table 1, a summary of the description of the datasets is given. The summary includes the number of attributes, the number of instances, the Imbalance Ratio (IR) among majority and minority classes, and the number of classes. For validation purposes, we used the Leave One Out validation (LOO).

3.2 Performance Measures and Statistical Tests

Imbalanced datasets, or otherwise known as class imbalance problems, manifest their presence when one or more class is underrepresented in the dataset. Some classic performance measures produce a bias towards the majority class in this type of imbalanced datasets. Therefore, these measures become inadequate for classification. To evaluate the performance in this type of datasets, the use of the average True Positive Rate for each class have been recently suggested [19].

		Real Class	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Fig. 1. Confusion matrix with data distributed in two classes.

The True Positive Rate (TPR) considers the total positive instances correctly classified in relation to the total instances of the positive class for problems of two classes (Fig. 1), in terms of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) is expressed in equation (1). Also, the True Positive Rate is known as recall or sensitivity.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

For a problem with k classes, the sensitivity of the classification in class j calculates the probability of correctly classifying an instance of class j, taking the total of correctly classified instances of class j in relation with the total instances of class j. For the calculation of the sensitivity of the classification is shown in the following equation:

$$S_j = \frac{n_j}{t_j} \tag{2}$$

where S_j is the sensitivity (also recall or True Positive Rate) of the classification of class j, n_j is the number of instances correctly classified and t_j is the total number of instances in class j.

The minimum classification sensitivity is a measure that allows us to handle multiple classes, and is expressed as follows [20]:

$$Minimum = \{S_j\} \tag{3}$$

Although minimum classification sensitivity only considers the lowest of all the classification sensitivities calculated. On the other hand, the average classification sensitivity per class [21] is a measure that gives the same weight to each of the classes, regardless of the number of instances that each class has. For this reason, this paper will use this measure of performance, and is denoted as follows:

$$Average\ sensitivity = \frac{1}{k} \sum_{i=1}^k S_j \tag{4}$$

where k is the class total and S_j is the classification sensitivity for j-th class. This measure of performance allows us to consider all classes without there being a preferential bias to any class.

Below we provide an example with data distributed in three classes, and where the average classification sensitivity and the minimum classification sensitivity are calculated:

		Real Class		
		X	Y	Z
Predicted Class	X	3	3	9
	Y	8	8	2
	Z	6	5	7

Fig. 2. An example with data distributed in three classes.

$$S_X = \frac{3}{15} = 0.2, S_Y = \frac{8}{18} = 0.444, S_Z = \frac{7}{18} = 0.388,$$

$$Average = \frac{0.2 + 0.444 + 0.388}{3} = 0.3444,$$

$$Minimum = \min\{0.2, 0.444, 0.388\} = 0.2.$$

In order to recognize which of the classification algorithms obtained the best experimental result, statistical hypothesis tests are used, which perform an analysis to evaluate whether there is or not a significant difference in the performance obtained by the models in the proposed datasets. For this paper the Friedman test, which is a non-parametric test, will be used, since it is highly recommended for the type of the analyzed data [22, 23].

Deepening, the Friedman [24] test can be viewed as an extension of the Wilcoxon test to include data recorded in more than two time periods or groups of three or more matched subjects. The test examines the ranges of the data generated in each time period to determine if the variables share the same continuous distribution of their origin.

3.3 Classification Algorithms

Some of the most recognized classification algorithms in the state of the art are: the decision trees, Nearest Neighbor, Naïve Bayes, Random Forest, Support Vector Machines (SVMs) and Sequential Minimal Optimization, which is a case of SVMs. In the next section, these algorithms were used to evaluate the datasets proposed.

Starting with the descriptions, a decision tree (DT) [25] is a map of the possible outcomes of a series of related decisions. The approach based on decision trees is a method commonly used in machine learning. The objective of this approach is to create a model that predicts the value of a target variable based on various input variables. Learning based on this type of approach is the construction of a decision tree from training columns, each labeled with its corresponding class. Some of the most used DT is C4.5, which generates a decision tree from the data through partitions performed recursively. C4.5 is a decision tree which is an extension of the ID3 algorithm.

Random Forests [26] are a combination of decision tree predictors, so that each tree depends on the values of a random vector sampled independently and with the same distribution. They are efficient for large data sets and can handle a large number of features.

The Statistical-Probabilistic approach is often based on the Bayes theorem, from which different methods have been inspired, such as Naïve Bayes [25]. The Bayes theorem tries to obtain the posterior probability in relation to the belonging of an instance to a certain class. For this, a priori probability of a given class is evaluated against the other classes.

Nearest Neighbor (NN) [25] or k nearest neighbor algorithm, is a supervised classification method based on metrics, whose training phase consists of storing the vectors of the training set. While in the test phase the distance between the stored vectors and the test vector is calculated, finally the "k" closest instances are selected; the class that is repeated more times is selected.

Support Vector Machines address the principle of risk minimization structure, that is why we can provide a generalized performance independent of the distribution of patterns. The central idea of the Support Vector Machines is the adjustment of a discrimination function that optimally uses the information of the patterns that separate the classes [25].

Finally, SMO [27] or Sequential minimal optimization, was an implementation of SVMs by John Platt. This model solves the problem of quadratic programming that had the Support Vector Machines during the training process. Another advantage that SMO

has is that the amount of memory used in training is linear, so it allows the use of large amounts of data.

4 Experimental Results and Discussion

In this section we present the experimental results of the classification models described above in the microarrays data for the classification of different types of cancer. We include k nearest neighbor (kNN using $k = 3$), a decision tree algorithm (C4.5), a Random Forest with 10 base trees, a Support Vector Machine version (SVM with polynomial grade 3 base) and SMO algorithm with polynomial grade 1 base. We used the default parameter values offered in the WEKA software package.

For the division of data between the training set and the test for the classifiers, Leave-one-out cross-validation (LOO) was used.

The results obtained are shown in the table 2, highlighting with bold the best results.

Table 2. Average True Positive Rate obtained by the classification algorithms.

Datasets	C4.5	kNN	Naïve Bayes	Random Forest	SVM (polynomial)	SMO
Leukemina	0.936	0.981	0.863	0.818	1.00	1.00
Colon	0.755	0.741	0.811	0.859	0.869	0.834
Adenocarcinoma	0.479	0.526	0.541	0.500	0.700	0.684
Brain tumor	0.635	0.680	0.585	0.760	0.810	0.855
Breast cancer 1	0.454	0.526	0.579	0.594	0.587	0.632
Breast cancer 2	0.413	0.588	0.653	0.537	0.544	0.595
Lymphoma	0.727	0.992	0.858	0.925	0.962	1.00
NCI	0.329	0.625	0.455	0.534	0.492	0.644
Prostate cancer	0.862	0.862	0.635	0.853	0.902	0.882
SRBCT	0.781	0.869	0.947	0.955	0.989	0.989
Times Best	0	0	1	0	5	6

The Friedman test was carried out in order to find the best classification method in a more appropriate way.

Table 3. Average Rankings of the algorithms (Friedman).

No.	Algorithm	Ranking
1	SMO	1.6
2	SVM	2.2
3	kNN	3.75
4	Random Forest	3.8
5	Naïve Bayes	4.3
6	C4.5	5.35

Table 3 shows the average of ranges obtained by each classifier in the Friedman test, where the best model was SMO. The Friedman test gives a probability of $p = 0.000052$, lower than the 0.05 significance value. But to validate if this model has a significant statistical difference, p-values are obtained by applying the Post Hoc comparison, where the Holm's procedure rejects those hypotheses that an unadjusted value of $p \leq 0.05$.

Table 4. Post Hoc comparison Table (Friedman).

No.	Algorithm	p
1	SVM	0.473289
2	kNN	0.010177
3	Random Forest	0.008551
4	Naïve Bayes	0.00125
5	C4.5	0.000007

In Table 4, the p-values that are less than or equal to 0.05 are marked (bold letters), which are the values that the test rejects the hypothesis, in other words, that there is a significant difference of the best ranked method (SMO) with respect to kNN, Random Forest, Naïve Bayes and C4.5 models. In contrast, the SVM model has a p-value greater than 0.5, so the test does not reject the null hypothesis. Therefore, SMO has no significant difference with SVM.

5 Conclusions and Future Work

In this document, we aim at evaluating several Machine Learning algorithms for breast cancer pre-diagnosis, based on microarray data. The evaluations made allow us to determine the best performing algorithms for this task.

The results that were obtained when analyzing the selected algorithms with microarray dataset, the SMO model was the one that obtained the best results. But when performing the Friedman test and the Holm post hoc test, it was shown that since there is no significant difference with the SVM model, it is just as feasible to use SVM to obtain good results. Considering these results, the best classifiers for this type of datasets were the SMO and SVM models.

As a future work, we would propose an algorithm related to the models that had better results (SVM and SMO) or a new algorithm different from those already evaluated, and that competes with the evaluated classifiers, obtaining a better result and obtaining a significant difference with the others. This new algorithm must be proposed to manage dataset with a large amount of data, such as microarrays data.

Acknowledgements. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, CIDETEC, and CIC), the CONACyT, and SNI for their economic support to develop this work

References

1. Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16(10), 906–914 (2000)
2. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439), 531–537 (1999)

3. Quackenbush, J.: Computational analysis of microarray data. *Nature Reviews Genetics* 2, 418–427 (2001)
4. Bhaskar, H., Hoyle, D.C., Singh, S.: Machine learning in bioinformatics: A brief survey and recommendations for practitioners. *Computers in biology and medicine* 36(10), 1104–1125 (2006)
5. Díaz-Uriarte, R., De Andres, S.A.: Gene selection and classification of microarray data using random forest. *BMC bioinformatics* 7(1), 3 (2006)
6. Birnbaum, D.J., Finetti, P., Lopresti, A., Gilabert, M., Poizat, F., Raoul, J.L., Delperio, J.R., Moutardier, V., Birnbaum, D., Mamessier, E., Bertucci, F.: A 25-gene classifier predicts overall survival in resectable pancreatic cancer. *BMC Medicine* 15(1), 170 (2017)
7. Dwivedi, A.K.: Artificial neural network model for effective cancer classification using microarray gene expression data. *Neural Computing and Applications* 29(12), 1545–1554 (2018)
8. Korucuoglu, M., Isci, S., Ozgur A., Otu H.H.: Bayesian Pathway Analysis of Cancer Microarray Data. *PLOS ONE* 9(7), e102803 (2014)
9. Tan, A.C., Naiman, D.Q., Xu, L., Winslow, R.L., Geman, D.: Simple decision rules for classifying human cancers from gene expression profiles. *Bioinformatics* 21, 3896–3904 (2005)
10. Cho, S.B., Won, H.H.: Cancer classification using ensemble of neural networks with multiple significant gene subsets. *Applied Intelligence* 26(3), 1573–1597 (2007)
11. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine learning* 46(1-3), 389–422 (2002)
12. Hu, Z., Fan, C., Oh, D.S., Marron, J.S., He, X., Qaqish, B.F., Livasy, C., Carey, L.A., Reynolds, E., Dressler, L., Nobel, A., Parker, J., Ewend, M.G., Sawyer, L.R., Wu, J., Liu, Y., Nanda, R., Tretiakova, M., Orrico, A.R., Dreher, D., Palazzo, J.P., Perreard, L., Nelson, E., Mone, M., Hansen, H., Mullins, M., Quackenbush, J.F., Ellis, M.J., Olopade, O.I., Bernard, P.S., Perou, C.M.: The molecular portraits of breast tumors are conserved across microarray platforms. *BMC genomics* 7(1), 96 (2006)
13. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition* 39(12), 2383–2392 (2006)
14. Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* 97(457), 77–87 (2002)
15. Dettling, M., Buhlmann, P.: Supervised clustering of genes. *Genome Biology* 3 (12), 0069.1–0069.15 (2002)
16. Ramaswamy, S., Ross, K.N., Lander, E.S. Golub, T.R.: A molecular signature of metastasis in primary solid tumors. *Nature Genetics* 33, 49–54 (2003)
17. Díaz-Uriarte, R., De Andres, S.A.: Gene selection and classification of microarray data using random forest. *BMC bioinformatics* 7(1), 3 (2006)
18. Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S.S., de Rijn, M.V., Waltham, M., Pergamenschikov, A., Lee, J.C., Lashkari, D., Shalon, D., Myers, T.G., Weinstein, J.N., Botstein, D., Brown, P.O.: Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics* 24 (3), 227–235 (2000)
19. Fernández, A., López, V., Galar, M., Del Jesus, M.J., Herrera, F.: Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-based systems* 42, 97–110 (2013)
20. Fazekas, M.: Analysing Data of Childhood Acute Lymphoid Leukaemia by Seasonal Time Series Methods. *JUCS (Journal for Universal Computer Science)* 12(9), 1190–1195 (2006)
21. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences* 250, 113–141 (2013)

22. Demšar, J.: Statistical comparisons of classifiers over multiple datasets. *Journal of Machine Learning Research* 7, 1–30 (2010)
23. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences* 180(10), 2044–2064 (2010)
24. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics* 11, 86–92 (1940)
25. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern classification*. Wiley, New York (1973)
26. Breiman, L.: Random forests. *Machine learning* 45(1), 5–32 (2001)
27. Platt, J.: *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. Microsoft Research. Technical Report MSR-TR-98-14 (1998)

Procesador dedicado para el tratamiento de datos en formato de punto flotante

Úrsula Samantha Morales-Rodríguez¹, Gelacio Castillo-Cabrera²,
Herón Molina-Lozano¹

¹ Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México

² Instituto Politécnico Nacional, Escuela Superior de Cómputo,
Ciudad de México, México
smoralesr-ipn@hotmail.com, hmolina@cic.ipn.mx,
gcastilloc@ipn.mx

Resumen. El desarrollo de hardware en el sector mercantil es considerablemente bajo respecto al desarrollo de software a nivel mundial, cabe mencionar que México no cuenta con el impulso adecuado para implementar nuevas arquitecturas dedicadas a la resolución de problemas específicos en Hardware para el perfeccionamiento de procesos o análisis de datos de una manera óptima. Consecuentemente y en pro de mejorar las estadísticas, el presente artículo muestra el desarrollo y la implementación de un procesador dedicado basado en la arquitectura RISC que permite el procesamiento de datos en formato de punto flotante consumiendo un solo ciclo de reloj por instrucción, ya que cuenta con una Unidad de Punto Flotante (UPF) dedicada que permite la ejecución de operaciones entre números enteros y decimales siguiendo el estándar IEEE-754 para su procesamiento.

Palabras clave: procesador dedicado. IEEE-754. UPF. punto flotante.

Dedicated Processor for Data Processing in Floating Point Format

Abstract. The development of Hardware in the commercial sector is considerably low with respect to the development of Software worldwide, it is worth mentioning that Mexico does not have the appropriate impulse to implement new architectures dedicated to the resolution of specific problems in Hardware for the improvement of processes or analysis of data in an optimal way. Consequently, and in order to improve statistics, this article shows the development and implementation of a dedicated processor based on the RISC architecture that allows data processing in floating point format consuming a single clock cycle per instruction, since it counts with a dedicated Floating Point Unit (UPF) that allows the execution of operations between whole numbers and decimals following the IEEE-754 standard for processing.

Keywords: dedicated processor. IEEE-754. UPF floating point.

1. Introducción

El desarrollo de nuevas tecnologías en el ámbito computacional ha incrementado debido al impulso de dos grandes actividades; el desarrollo de *Hardware* y el desarrollo de *Software*. En el sector industrial, el desarrollo de hardware es considerablemente bajo siendo ca. 19%, en comparación al desarrollo de software que abarca ca. 81% del mercado a nivel mundial [1], dichos porcentajes muestran limitaciones para implementar nuevas tecnologías por la falta de hardware. Estadísticamente, el continente europeo cuenta con un mayor desempeño desarrollando hardware en comparación con el continente americano [2], consecuentemente México carece del impulso adecuado y necesario para implementar nuevas arquitecturas dedicadas a la resolución de problemas específicos en Hardware que ayuden al perfeccionamiento de procesos o análisis de datos de una manera óptima y precisa.

Es necesario que desde el nivel académico se desarrolle hardware para el funcionamiento eficiente de diversas aplicaciones de impacto e interés en el sector industrial, significando que las tareas se realicen de una manera más rápida, con un menor consumo de potencia y/o minimizando el espacio en cuanto a recursos, entendiéndose por hardware a la arquitectura física para el procesamiento de la información, es decir, un procesador [3].

El presente trabajo comprende el desarrollo, desde el diseño hasta la simulación de un procesador dedicado para la ejecución de tareas específicas realizando operaciones en punto flotante y pretende impulsar un amplio panorama de nuevas tecnologías en diseño y desarrollo de hardware, y por consecuencia, abrir camino hacia una ilimitada fuente de ideas para el desarrollo de software.

La unidad lógica aritmética del procesador diseñado realiza operaciones en formato de punto flotante, siendo ésta su principal característica, cuya ventaja es la precisión de los resultados en comparación de la unidad lógica aritmética simple. Actualmente, las operaciones con punto flotante son implementadas en distintas aplicaciones; de telecomunicaciones, control industrial, procesamiento de señales, entre otras. A pesar de la incorporación de las computadoras en los campos mencionados, la falta de sistemas dedicados que permitan la realización de diversas tareas específicas sin desperdicio de recursos y eficiencia en su desempeño prevalece.

En la actualidad existen diversos procesadores dedicados desarrollados a nivel académico; por citar dos de ellos se aprecia un proyecto publicado en 2013 a nivel internacional en el congreso “International Conference on Advanced Electronic Systems” (ICAES) desarrollado en la universidad VIT en India [4] el cual presenta un co-procesador con una Unidad Lógica Aritmética de arquitectura segmentada de punto flotante para minimizar la potencia e incrementar la frecuencia de operación con la intención de mejorar el rendimiento del diseño usando Verilog HDL, sin embargo utiliza más de un ciclo de reloj para poder ejecutar las operaciones básicas, mientras que el segundo se trata de un proyecto académico desarrollado en la Escuela Superior de Computo del Instituto Politécnico Nacional [5] donde se implementa un microprocesador en VHDL de 16 bits de arquitectura RISC tipo MIPS con arquitectura Harvard y una Unidad Lógica aritmética simple con esquema de acarreo anticipado por propagación que no permite el procesamiento de datos en punto flotante, no obstante cada instrucción ejecutada se lleva a cabo en un ciclo de reloj.

El presente artículo muestra el diseño y el desempeño de un procesador dedicado de 32 bits de arquitectura tipo Harvard con una Unidad de Punto Flotante (UPF) diseñada únicamente para permitir el procesamiento de datos en formato punto flotante de precisión simple de acuerdo al estándar IEEE-754 [6]; se realiza el diseño implementando una filosofía orientada a procesadores RISC basados en MIPS, modelando la arquitectura mediante lenguaje de descripción de Hardware VHDL y mostrando como característica principal la ejecución de una instrucción por cada ciclo de reloj, es decir, posee una arquitectura monociclo.

2. Diseño del procesador

Se diseñó un procesador dedicado de 32 bits basado en la arquitectura RISC tipo MIPS, cuyas principales características son: los datos procesados únicamente serán del tipo Punto Flotante, únicamente se realizarán operaciones sobre registros, cuenta con una memoria de programa y una memoria para datos, así como con un banco de registros, solo contiene la Unidad de Punto Flotante (UPF) para realizar operaciones, un Contador de Direcciones, el cual indica el conteo en la dirección al ejecutarse un programa y un Contador de Programa (CP) para indicar direcciones de memoria. La unidad de control es la encargada de controlar las señales destinadas al buen funcionamiento de los componentes de acuerdo al tipo de instrucción implementada. La Fig. 1 Muestra el diagrama a bloques del procesador dedicado, donde se puede apreciar la arquitectura y la interconexión de cada componente.

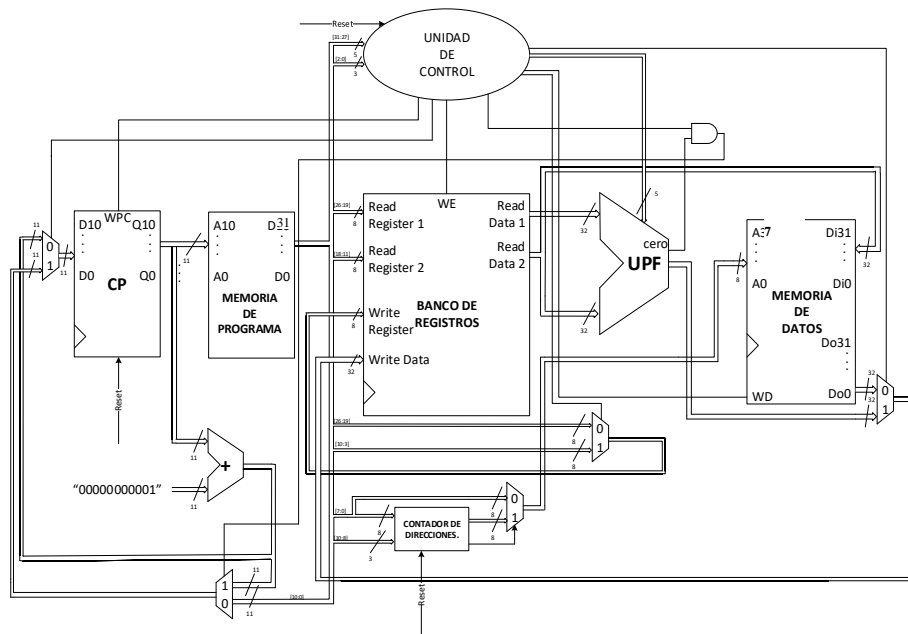


Fig. 1. Diagrama general a bloques del procesador dedicado de 32 bits en punto flotante.

2.1. Formato de instrucciones

Para la definición del formato de instrucciones, se considera la representación de las instrucciones del procesador formadas por una cadena de 32 bits cada una y el procesamiento del tipo de dato en formato de punto flotante según el estándar IEEE-754 de precisión sencilla, estableciendo campos para cada tipo de instrucción de acuerdo con la Tabla 1.

Las instrucciones tipo R son instrucciones de tipo aritmético-lógicas; mediante el código de operación (op) indicado realizan una operación en la UPF sobre los contenidos en los registros fuentes 1 y 2 (rs, rt) escribiendo el resultado sobre el registro final (rd) apoyándose del campo resultante (cr) para la definición del tipo de operación. Las instrucciones tipo Jump y Branch son instrucciones de salto; de tipo incondicional, con 5 bits para el código de operación y los 11 bits menos significativos son para el campo de dirección (dir), o de tipo condicional, donde se debe especificar dos campos en los operandos (rs, rt) para realizar la diferencia, además de la dirección de salto, respectivamente. Las instrucciones de transferencia de datos pueden ser de carga o almacenamiento; el formato de la instrucción carga/almacenamiento cuenta con su respectivo código de operación (op), un registro fuente o destino (rd, rs) y la dirección de memoria (dir).

Tabla 1. Descripción de los tipos de formato de instrucción con sus respectivas definiciones.

Tipo de instrucción		Formato de instrucción					Definiciones	
Sobre Registro		op	rs	rt	rd	cr	op	Código de operación
		5 bits	8 bits	8 bits	8 bits	3 bits	rs	Registro 1/ almacenamiento
Salto	Condicional	op	rs	rt	dir		rt	Registro 2
		5 bits	8 bits	8 bits	11 bits		rd	Registro final / carga
	Incondicional	op	s/u	s/u	dir		cr	Campo resultante
		5 bits	8 bits	8 bits	11 bits		s/u	Sin uso
Transf. de datos	Carga	op	rd	s/u	dir		dir	Dirección de memoria
		5 bits	8 bits	8 bits	11 bits			
	Almacenamiento	op	s/u	rs	dirección			
		5 bits	8 bits	8 bits	11 bits			

2.2. Set de instrucciones

El procesador dedicado maneja un total de 9 operaciones en el set de instrucciones de tipo salto, aritmético-lógicas y transferencia de datos; se pueden consultar en la Tabla 2. Cabe mencionar que cada instrucción consume un ciclo de reloj en el procesador y todos los operandos utilizarán su respectiva representación en formato punto flotante para ser procesados de acuerdo al estándar IEEE-754.

Tabla 2. Set de instrucciones del procesador dedicado.

Tipo	Codop	Instrucción	Nemónico	Ejemplo	Significado
Store	00000	Carga	LW	LW rd, dir	rd = Mem [dir]
Store	10000	Almacenamiento	SW	SW rs, dir	Mem[dir] = rs
R	00001	Suma	ADD	add rd,rs,rt	rd = rs + rt
R	00010	Resta	SUB	sub rd,rs,rt	rd = rs - rt
R	00011	Multiplicación	MULT	mult rd,rs,rt	rd = rs * rt
R	00100	División	DIV	div rd,rs,rt	rd = rs / rt
Branch	10010	Salto si es diferente	BNEQ	BNEQ dir	If(rs!=rt) go to Dir
Jump	10001	Salto	B	B dir	PC = Dir
	11111	NOP	NOP	NOP	--

2.3. Formato del tipo de dato

De acuerdo con la intención que el estándar IEEE-754 refiere, donde la implementación de cualquier sistema en punto flotante puede ser completamente desarrollada a nivel software, a nivel hardware o cualquier combinación entre estos dos niveles, el formato del tipo de dato a ser procesado es el formato de precisión simple que sigue el estándar, donde la representación de un número en coma flotante contempla una parte entera y una parte decimal. Siguiendo el estándar, una vez aplicada la conversión de dicha representación al formato de precisión simple [6], el cual posee un bus de 32 bits dividido en tres campos reservados, el primer campo cuenta con el bit más significativo (bit31) indicando el signo del número a ser representado, un segundo campo que reserva 8 bits para representar el exponente (bit30 ... bit23) y un último campo con 23 bits para representar la mantisa (bit22 ... bit0), el formato final puede ser apreciado en la Fig. 2.

$$(1.853515625 \times 2^6) \times (-1) = -118.625$$

Bit de Signo (S)	Exponente (E)	Mantisa (M)
1	1000 0101	11011010100000000000000

Fig. 2 Representación científica del número -118.625 y su respectiva representación en formato de precisión simple de punto flotante según el estándar IEEE-754 [7].

2.4. Componentes del procesador

El procesador cuenta con siete componentes fundamentales para su funcionamiento: el contador de programa, contador de direcciones, la memoria de programa, el banco de registros, la memoria de datos, la unidad de control y la unidad de punto flotante (UPF).

Contador de programa: registro que contiene la dirección de la consecuente instrucción a ejecutar obtenida de la memoria de programa, dicho registro será incrementado automáticamente por cada ciclo de instrucción.

Contador de direcciones: permite el incremento de la dirección contigua a ser leída por el contador de programa según se trate de la instrucción a ser ejecutada de acuerdo al formato de instrucciones.

Memoria de programa: memoria de solo lectura; las instrucciones del programa a procesar serán almacenadas en esta memoria. Su capacidad de memoria es de 2K x 4B, la instrucción leída de la memoria es dirigida hacia la unidad de control, el banco de registros y el contador de direcciones; cuenta con un bus de direcciones y un bus de datos.

Banco de registros: conformado por 256 registros de 32 bits cada uno; cuenta con dos puertos de lectura y un solo puerto de escritura, los puertos de lectura con 8 bits a la entrada, Read Register1 y Read Register2, haciendo posible direccionar hasta 256 localidades del respectivo banco de registros, cabe mencionar que siguiendo la implementación de una arquitectura tipo MIPS donde el banco de registros direcciona hasta 32 localidades, el presente diseño considera la implementación del total de localidades disponibles a partir de la capacidad de direccionamiento disponible sin descartar la implementación de una menor cantidad de registros basándonos en la arquitectura MIPS donde únicamente se podrían considerar los 4 bits menos

significativos de los buses de entrada en los puerto de lectura; dichos puertos son leídos a través de las salidas Read Data1 y Read Data2, respectivamente. El puerto de escritura toma 8 bits de dirección a la entrada de Write Register, así como 32 bits de datos para la entrada de Write Data. Posee un bit de control denominado WE, el cual, si se encuentra posicionado en alto, el banco de registros es capaz de escribir los datos recibidos (Write Data) en el registro especificado (Write Register), en caso contrario es de solo lectura sobre los registros fuente.

Memoria de datos: una memoria de lectura/escritura, su capacidad de memoria es de 256 x 32 bits; debido a su ancho de palabra permite almacenar datos en formato de punto flotante con precisión simple. Maneja dos buses de datos de 32 bits para entrada y salida, así como un bus de direccionamiento de 8 bits; considera una menor capacidad de almacenamiento en comparación con la memoria de programa, ya que el diseño del presente procesador prospecta el tratado de una gran cantidad de instrucciones debido a la naturaleza de su arquitectura.

Unidad de control: Para implementar la unidad de control como una máquina de estados finita, primero se asignó un número a cada uno de los 9 estados, en este caso, se usó un enumerado simple secuencial; el diagrama de la máquina de estados finita implementada se aprecia en la Fig. 3. conteniendo nueve estados con su debida representación en binario (4 bits) para codificar el número de estado representado.

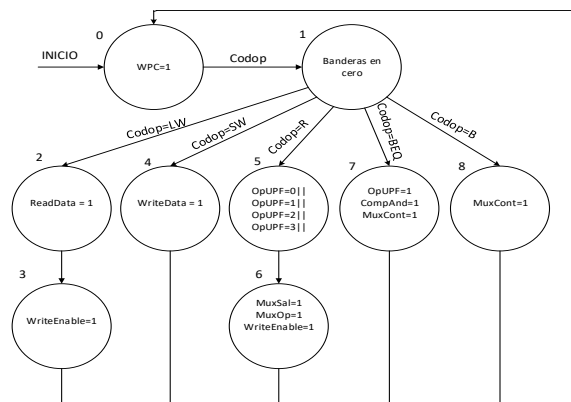


Fig. 3 Diagrama de la máquina de estados finita para la unidad de control.

La **Tabla 3** muestra de manera general las banderas de activación utilizadas por la unidad de control por cada camino de datos respectivo de cada instrucción implementada en el procesador.

Tabla 3. Bits de entradas y salidas manejados por la unidad de control.

Control	Nombre de la señal	Carga	Almacenamiento	Tipo R				Beq	B
				ADD	SUB	MULT	DIV		
Entradas	Codop(4)	0	1	0	0	0	0	1	1
	Codop(3)	0	0	0	0	0	0	0	0
	Codop(2)	0	0	0	0	0	1	0	0
	Codop(1)	0	0	0	1	1	0	1	0
	Codop(0)	0	0	0	1	0	1	0	0
Salidas	MuxCont	0	0	0	0	0	0	1	0
	WriteEnable	1	0	1	1	1	1	0	0

Control	Nombre de la señal	Carga	Almacenamiento	Tipo R				Beq	B
				ADD	SUB	MULT	DIV		
	MuxOp	1	0	1	1	1	1	0	0
	ReadData	0	0	0	0	0	0	0	0
	WriteData	1	1	0	0	0	0	0	0
	MuxSal	0	0	1	1	1	1	0	0
	CompAnd	0	0	0	0	0	0	1	0
	MuxBranch	0	0	0	0	0	0	1	1
	WPC	1	1	1	1	1	1	1	1
	OpUPF(2)	0	0	0	0	0	0	0	0
	OpUPF(1)	0	0	0	0	1	1	1	0
	OpUPF(0)	0	0	0	1	0	1	0	0

Unidad de Punto Flotante [7]: se implementó una unidad de punto flotante que permite el procesamiento únicamente de números con el formato de punto flotante de precisión simple del estándar IEEE-754. Se trata de una unidad dedicada que realiza el procesamiento de los operandos en un solo ciclo de reloj, sacrificando espacio debido a una mayor densidad en cuanto a recursos de hardware, pero considerando una mejora en el tiempo de respuesta, la arquitectura se observa en la Fig.4.

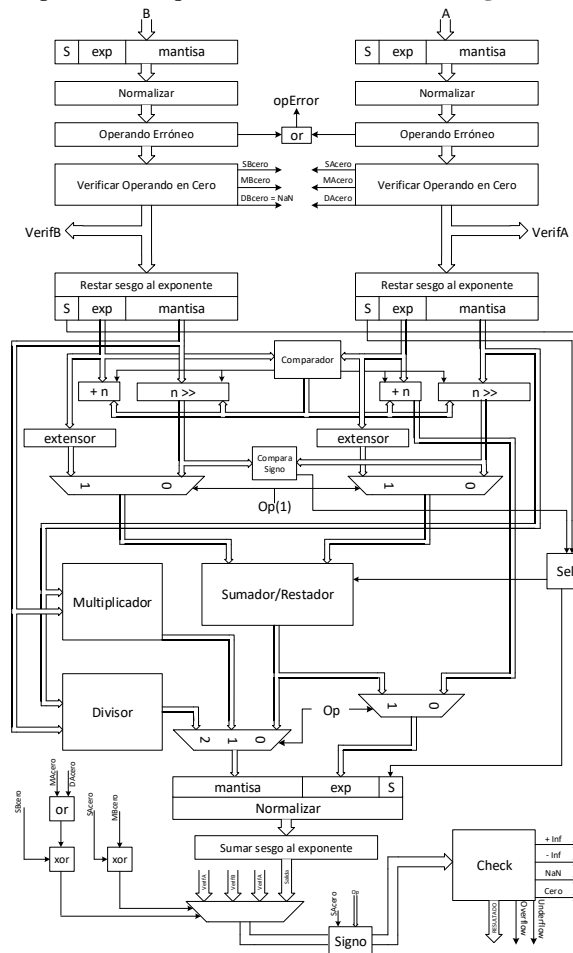


Fig. 4. Diagrama general de la unidad de punto flotante para el procesamiento de datos en formato de precisión simple de acuerdo al estándar IEEE-754.

3. Campo de aplicación

Para visualizar la operatividad del procesador, se incluyó en la memoria de programa una serie de instrucciones que permiten conocer el resultado de procesar parte del funcionamiento de la primera iteración en la etapa de entrenamiento de la red neuronal ADALINE.

Una neurona ADALINE es una neurona simple de una unidad, recibe entradas de varias unidades y una entrada de una “unidad” la cual siempre tiene el valor de +1 con el objetivo de entrenar el peso del sesgo por el mismo proceso mediante la regla delta, usado para poder actualizar los pesos de cada entrada ponderada [8].

El algoritmo de entrenamiento de una neurona ADALINE es el siguiente:

- Paso 0. Inicializar los pesos
- Paso 1. Mientras la condición de paro sea falsa, realiza los pasos 2-6.
- Paso 2. Colocar la activación de las unidades de entrada
- Paso 3. Calcular la entrada de la red hacia la salida de la unidad dada por:

$$y_{in} = b + \sum_1^i x_i w_i.$$

- Paso 4. Actualiza sesgo y pesos
- Paso 5. Verificar la condición de paro.

En términos generales, el procesador se encarga de ponderar los pesos por los valores de las señales, realizar la suma acumulatoria y finalmente agregar el sesgo de la neurona, es decir, realiza el paso 3 del algoritmo de entrenamiento de la red neuronal con los valores que se muestran en la **Fig. 5**.

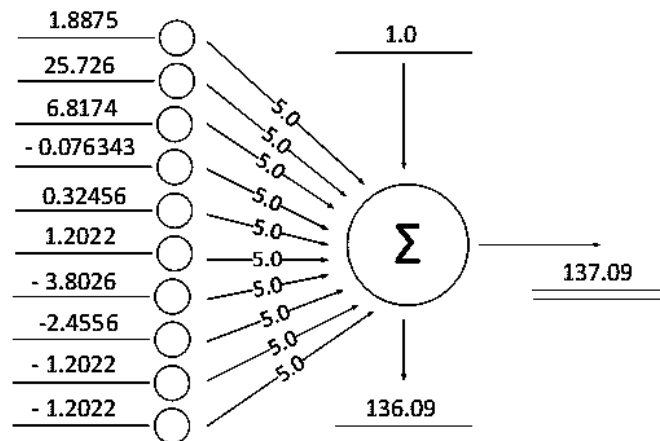


Fig. 5. Neurona ADALINE con valores numéricos.

El listado de instrucciones incluido en la memoria de programa del procesador es el mostrado en la Fig. 6.

```

LW R0, x"00"      ; R0= MEM[0] – Carga del número de muestras al registro 0
LW R1, x"01"      ; R1= MEM[1] – Carga el número uno al registro 1
LW R2, x"02"      ; R2= MEM[2] – Carga el número cero al registro 2
LW R6, x"02"      ; R6= MEM[2] – Carga el número cero al registro 6

CICLO:
LW R4, x"0D"      ; R4= MEM[13] – Carga el valor del peso
LW R5, x"03"      ; R5= MEM[3] -- Carga el valor de la señal
MULT R5, R4, R5   ; R5 = R4 * R5 – Multiplica el peso por la señal
ADD R6, R5, R6    ; R6 = R5 + R6 – Realiza la suma acumulatoria.
SUB R0, R1, R0    ; R0 = R1 – R0 -- Resta el número de muestras menos uno
BNEQ R0, R2, x"04" ; if (R2!=R0) go to CICLO – Si el número de iteraciones
                  ; es distinto al número de muestras, salta a la etiqueta CICLO,
                  ; en caso contrario, continúa.

ADD R6, R1, R6    ; R6 = R1 + R6 – Suma el sesgo a la suma acumulatoria.
SW R6, x"FF"     ; MEM[255] = R6 – Almacena en memoria el valor del registro 6.
LW R7, x"FF"     ; R7 = MEM[255] – Carga el resultado final sobre el registro 7.
B x"0C"          ; PC = 12 – Brinca a la dirección 12
    
```

Fig. 6 Programa detallado embebido en la memoria de programa para realizar el procesamiento del paso 3 del algoritmo de entrenamiento de la neurona ADALINE.

4. Resultados

Los resultados obtenidos en la simulación del funcionamiento del procesador con el programa ya embebido en la memoria utilizando el software ACTIVE HDL fueron los mismos que los resultados esperados.

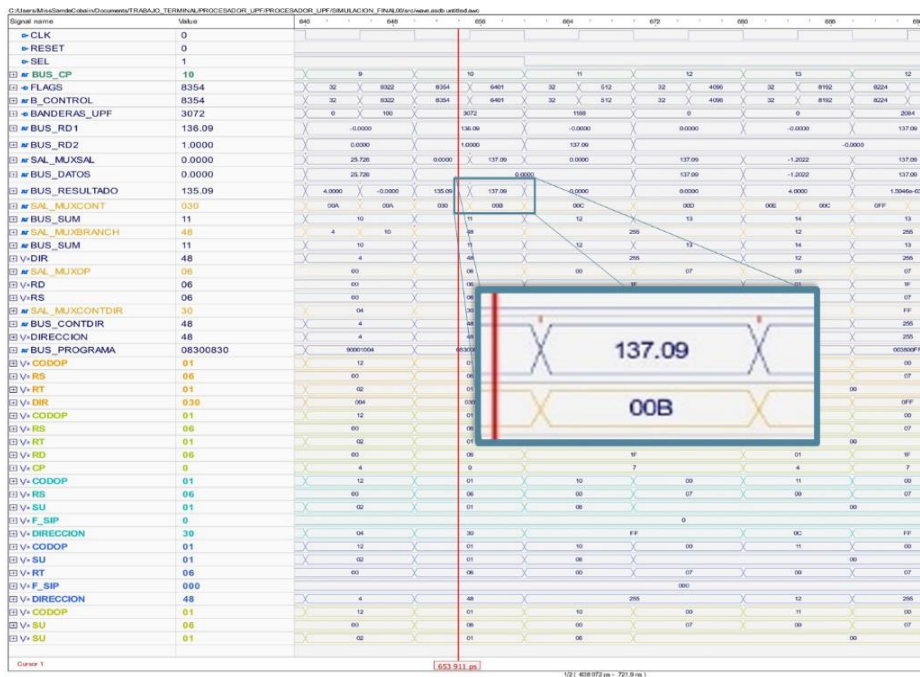


Fig. 7 Simulación del resultado final empleando el software ACTIVE HDL.

5. Conclusiones

Los resultados obtenidos en la simulación fueron los mismos resultados que los esperados, cabe mencionar que siguiendo el estándar IEEE-754 para tratar los números en formato de punto flotante, fue posible implementar en hardware una arquitectura dedicada capaz de procesar cualquier valor numérico existente en el formato presentado; estrictamente, el procesador únicamente maneja datos en punto flotante debido a que a los números enteros también los opera como datos en punto flotante.

El diseño del procesador dedicado de punto flotante fue basado en una arquitectura tipo MIPS, permitiendo ejecutar cada instrucción del mismo set de instrucciones en un solo ciclo de reloj procurando mejorar los tiempos de respuesta en comparación con otras arquitecturas, si bien, en ocasiones resulta complicado realizar comparativas entre arquitecturas dedicadas que sean capaces de procesar un mismo programa, no se descarta la elaboración de dicha comparativa como trabajo a futuro con la finalidad de evaluar el desempeño del presente procesador.

Como bien se sabe, la filosofía de la arquitectura de computadoras nos indica una proporcionalidad en relación densidad-tiempo, de modo que, al reducir el tiempo de procesamiento, se incrementa la densidad de nuestro diseño y ello se ve reflejado en la Unidad de Punto Flotante.

Hoy en día existen diversas arquitecturas dedicadas al procesamiento de datos y al análisis de señales, pero el procesador presentado es un procesador dedicado para diversas aplicaciones específicas, ya que podría ser implementado una en aparatos de tamaño mínimo que permitan la cancelación de ruido, el promedio de diversos valores para el diagnóstico y/o control de la hipertensión o la diabetes, la selección de una señal con mayor intensidad, entre otras.

Referencias

1. AESOFT, Asociación Ecuatoriana de Software: Estudio de mercado en el sector software y hardware en Ecuador. <http://www.aesoft.com.ec/> (septiembre, 2011)
2. AESOFT, Asociación Ecuatoriana de Software: Estudio de mercado en el sector software y hardware en Ecuador <http://www.aesoft.com.ec/> (septiembre, 2011)
3. Grediaga Olivo, A., Pérez Martínez, J.: Procesadores. Diseño de procesadores en VHDL. Publicaciones Universidad de Alicante Ed., Primera ed. España: Alicante. pp. 225 (2007)
4. Manisha S., Anita, A.: Design and Implementation of Single Precision Pipelined Floating Point Co-Processor. In: International Conference on Advanced Electronic Systems (ICAES), Nwe Delhi (2013)
5. García Ortega, V.H. et al.: Microprocesador didáctico de arquitectura RISC implementado en un FPGA. Universidad de Guadalajara, e-Gnosis, vol. Esp., pp 1–8 (2009)
6. Cornea: IEEE 754-2008 Decimal Floating-Point for Intel® Architecture Processors, In: 19th IEEE International Symposium on Computer Arithmetic (2009)
7. Morales Rodríguez, U.S.: Procesador de punto flotante para análisis de señales. Tesis (Ingeniería en Sistemas Computacionales). Ciudad de México, Instituto Politécnico Nacional, Escuela Superior de Cómputo, p. 202 (2016)
8. Fauset, L.: Fundamentals on Neural Networks, architectures, algorithms and applications. United States of America: Prentice Hall, Englewood Cliffs (1994)

Propuesta de sistema de gestión inteligente basado en IoT para hidroponía

Evelyn Gutierrez Leon¹, Jorge Erik Montiel Arguijo¹, Chadwick Carreto Arellano², Felipe Rolando Menchaca García¹

¹ Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco, Ciudad de México, México

² Instituto Politécnico Nacional, Escuela Superior de Cómputo, Ciudad de México, México
evelynleon@hotmail.com, jerikmontiel.isc@gmail.com,
ccarreto@ipn.mx, fmenchac@gmail.com

Resumen. Este artículo propone el diseño de un sistema de gestión inteligente basado en el paradigma de internet de las cosas (IoT) para el caso de uso de cultivos hidropónico. Se propone la arquitectura de red y diseño de prototipos enfocados a obtener la información en “tiempo real” de los parámetros clave que desarrollo de los cultivos hidropónicos de escala semi-comercial y de autoconsumo, así como la primera versión de la interfaz gráfica para dispositivos móviles que permitirán la gestión del cultivo y toma de decisiones. La premisa del proyecto es el desarrollo de una solución modular, escalable, de bajo costo, fácil instalación y uso.

Palabras clave: internet de las cosas, agricultura inteligente, IoT, redes inalámbricas.

Proposed Smart Management System Based on IoT for Hydroponics

Abstract. This article proposes the design of an intelligent management system based on the Internet of Things (IoT) paradigm for the case of hydroponic crop use. The network architecture and design of prototypes are proposed, focused on obtaining the information in “real time” of the key parameters that develop the hydroponic crops of semi-commercial and self-consumption scale, as well as the first version of the graphic interface for mobile devices that will allow crop management and decision making. The premise of the project is the development of a modular, scalable, low-cost solution, easy installation and use.

Keywords: internet of things, smart agriculture, IoT, wireless networks.

1. Introducción

La agricultura es la tercera actividad económica en el país, cerca de 22 millones de hectáreas son dedicadas al cultivo, de las cuales, el 75.67% implementan algún nivel de mecanización y de éstos 64.92% son cultivos de temporal [1].

De la población agrícola, aproximadamente el 35% de los agricultores se encuentra en el sector micro y pequeña empresa con lazos en el mercado interno y el 50% produce para autoconsumo. A pesar de existir diversas soluciones importadas para el campo agroalimentario tales como los invernaderos, equipos de monitoreo, de control climático y de riego, que ofrecen múltiples ventajas en la producción, éstos son enfocados a productores intensivos de escala industrial, los cuales requieren una alta inversión inicial y personal altamente calificado en la tecnología para su instalación, operación y mantenimiento, por lo cual, su implementación no ha sido generalizada para productores de los sectores pequeña, micro empresa y huertos de autoconsumo, quienes son más vulnerables ante la variabilidad y el cambio climático[2].

Dicho lo anterior, el principal reto en el campo agrícola es satisfacer la demanda de producción de alimentos manteniendo los niveles sustentables de los recursos naturales [3], por lo cual, la implementación de la ciencia y nuevas tecnologías en la agricultura es pieza clave para diseñar soluciones de bajo impacto ambiental. En donde, el desarrollo de soluciones basadas en Internet de las cosas (IoT) y la agricultura de precisión (AP), han creado un amplio panorama de desarrollo de dispositivos y equipos que satisfagan las necesidades de monitoreo y control de las diversas etapas dentro de la campaña de cultivo, las cuales han sido implementadas y adoptadas con gran ímpetu en múltiples países desarrollados [4].

La AP también es conocida como agricultura inteligente (o Smart Farming) dado que se caracteriza por el monitoreo constante a través del uso de diversas tecnologías de la Información y Comunicaciones (TIC) así como de la automatización, aunando el conocimiento científico y tecnológico con las metodologías agrícolas. En donde el papel de Internet de las cosas (IoT) ha sido ampliamente destacado [5 y 6].

El presente artículo se enfoca a describir el diseño del sistema enfocado en la primera etapa de diseño de dispositivos IoT para captación de la mayor cantidad de información posible para que en una futura fase, implementar algoritmos avanzados para procesar la información y realizar la toma de decisiones y autoconfiguración de los parámetros clave que influyen en el desarrollo del cultivo [7 y 8].

2. Diseño del sistema

Las características y requerimientos de los sistemas agrícolas semicomerciales y de autoconsumo suelen ser muy variables: la extensión del cultivo, la distribución de las plantas, tipo de sistema hidropónico. Por lo cual, para satisfacer las necesidades de monitoreo, control y gestión de los sistemas hidropónicos antes mencionados, es necesario que la solución cumpla las siguientes características: modular, flexible, configurable y de bajo costo.

Se propone un sistema en el cual, se conjugan conocimientos y desarrollos de la agricultura de precisión e Internet de las cosas, con la finalidad de reducir costos en el desarrollo de la solución, pero ofreciendo las capacidades de gestión de cultivos similares a las soluciones industriales y comerciales actualmente existentes (Fig. 1).



Fig. 1. Esquema de la solución de Agricultura inteligente basada en IoT y agricultura de precisión.

2.1. Arquitectura de red

Inicialmente la arquitectura de red fue basada en el modelo de referencia definido por el grupo de trabajo P2413 del IEEE-SA [9], al cual le fue agregado una cuarta capa (intermedia a la capa de red y la capa de aplicación) llamada capa de procesamiento de datos. (Fig. 2)

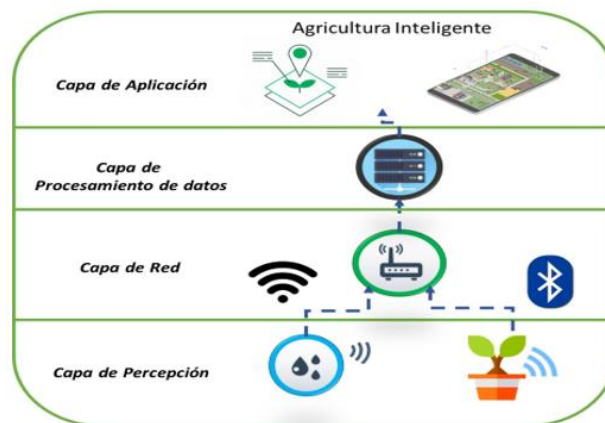


Fig. 2. Arquitectura de referencia IoT de cuatro capas para el sistema de gestión inteligente para cultivos hidropónicos "Pouani".

2.2. Capa de percepción

En la capa de percepción se encuentran los dispositivos capacitados por una serie de sensores y actuadores, dedicados a la tarea de adquirir información del entorno del cultivo dentro del invernadero y modificarla. Dentro de esta capa se definen dos tipos de dispositivos: Nodo Sensor y Nodo de Riego. El diseño de los nodos se caracteriza por constar de los bloques o unidades de alimentación (por ejemplo, baterías), percepción (sensores y actuadores), procesamiento (μC) y de comunicación (transceptores) cada uno (Fig. 3-5). Estos nodos a su vez son gestionados dentro del invernadero por el nodo Maestro.

Nodo Sensor

La tarea principal del Nodo Sensor (Fig. 3) es recolectar la información del entorno a través de diversos parámetros clave que influyen en el desarrollo del cultivo, tales como la Iluminación (lx), temperatura (°C), Humedad (%HR) y temperatura de la Solución Nutritiva (SN). Con la finalidad de obtener datos con mayor precisión, se pretende ubicar el nodo Sensor en el mismo módulo hidropónico en donde se coloca las plantas.

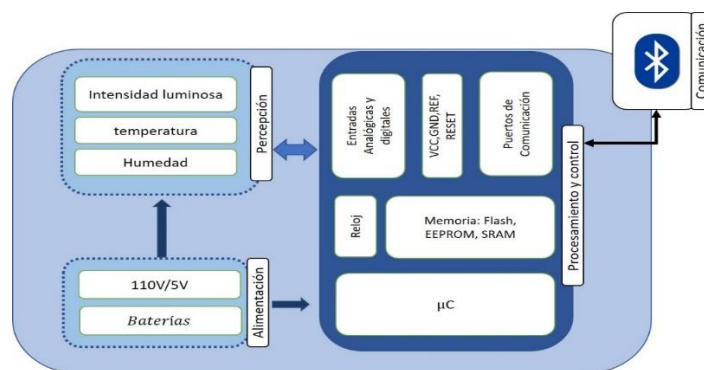


Fig. 3. Diagrama a bloques del Nodo Sensor.

Nodo de Riego

El nodo Riego (Fig. 4) obtiene información del estado de la Solución Nutritiva en el punto de concentración de esta, es decir, en el tanque de Riego en donde se realiza también el control del suministro de riego a cada módulo hidropónico a través de la activación y desactivación de las electroválvulas asignadas a cada uno.

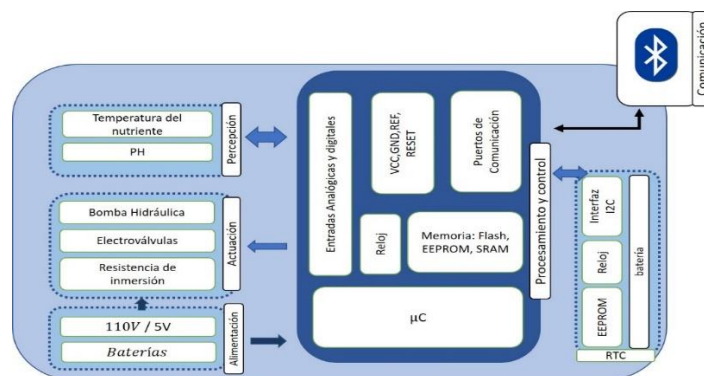


Fig. 4. Diagrama a bloques del Nodo Riego.

Dado que las condiciones de extensión dentro del invernadero pueden ser muy variadas, es decir, la cantidad de nodos dispuestos y la distancia de cada uno con respecto al servidor pueden ser diversos, una comunicación dentro del invernadero en topología

mallada pretende subsanar dicha heterogeneidad concentrando la información en primera instancia a un nodo maestro que posteriormente reenvíe los datos al servidor.

Capa de red

En la capa de red se soporta la conectividad de los equipos, en donde son aplicados diversos protocolos y tecnologías de comunicación orientados a establecer la conexión punto a punto de los dispositivos implicados en el proyecto para realizar la transferencia de datos entre ellos. Se definió un caso mínimo de topología de red, el cual cubre las necesidades de un huerto de autoconsumo o pequeño invernadero (Fig. 5) dividiendo la solución en dos áreas de aplicación: Intra Invernadero y Extra Invernadero.

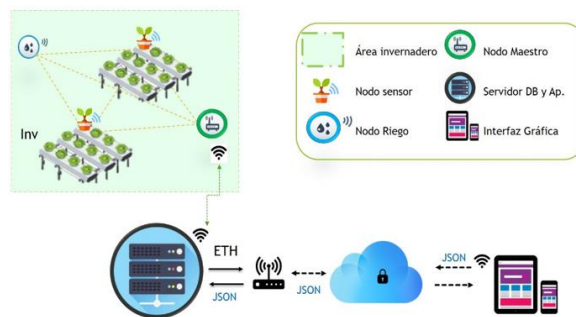


Fig. 5. Arquitectura general de red propuesta para el sistema de gestión inteligente para cultivos hidropónicos, caso mínimo.

En el área Intra-Invernadero se define la conexión entre los nodos por medio de una topología de red mallada con la finalidad de tomar ventaja de sus capacidades de área de cobertura, robustez y dinamismo. Dicha comunicación mallada es gestionada por el nodo Maestro (Figura 6), el cual asigna el identificador dinámico de los nodos, mantiene activa la red y concentra la información recaba en cada nodo.

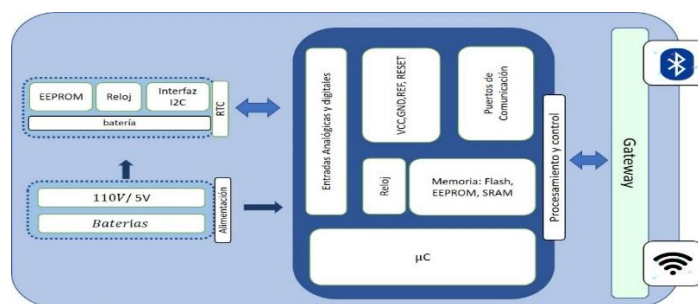


Fig. 6. Diagrama a bloques del nodo Maestro.

En el área Extra-Invernadero se considera la comunicación de la información concentrada por el nodo Maestro hacia el servidor y de la interfaz de usuario al servidor. Con la finalidad de reducir los requerimientos iniciales de instalación, el Servidor es configurado como punto de acceso inalámbrico (WAP) para proveer la comunicación

local de los nodos maestro y de la interfaz de usuario al servidor (fig. 7), en caso de que el usuario final cuente con la infraestructura, la conexión puede ser realizada a través del protocolo Ethernet y hacer uso de sus servicios (fig. 8).

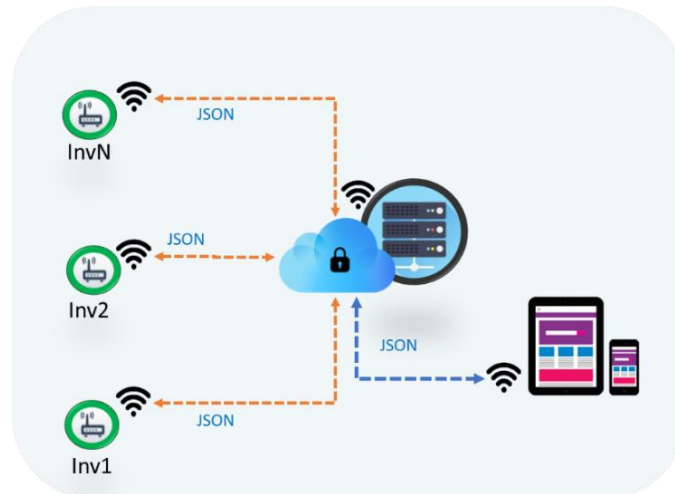


Fig. 7. Comunicación Extra Invernadero caso servidor configurado como WAP.

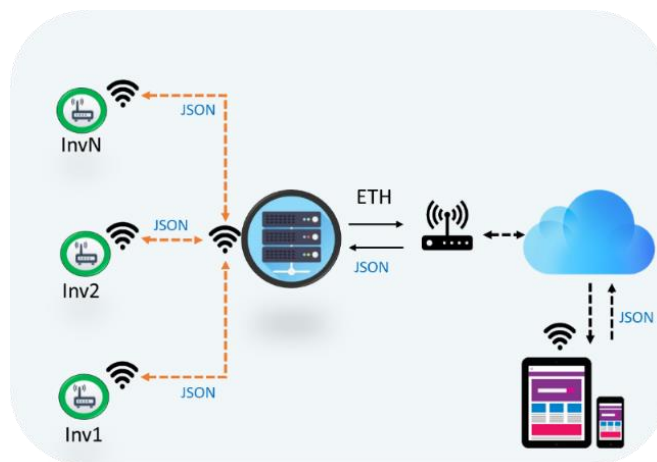


Fig. 8. Comunicación Extra Invernadero caso servidor conectado a la infraestructura del agricultor.

Dado que, el área destinada a un cultivo, el tipo de módulo hidropónico y la distribución de estos son muy diversos: entre agricultores, dentro de los mismos campos de cultivo o invernaderos, así como entre cada campaña agrícola; se definió una topología de red modular en diferentes etapas, para satisfacer las necesidades previamente mencionadas, permitiendo que el mismo modelo pueda ser implementado en diversos casos, desde huertos de autoconsumo hasta sistemas de cultivos desplegados en múltiples invernaderos. En la Fig. 9 se muestra la topología de red general de la solución.

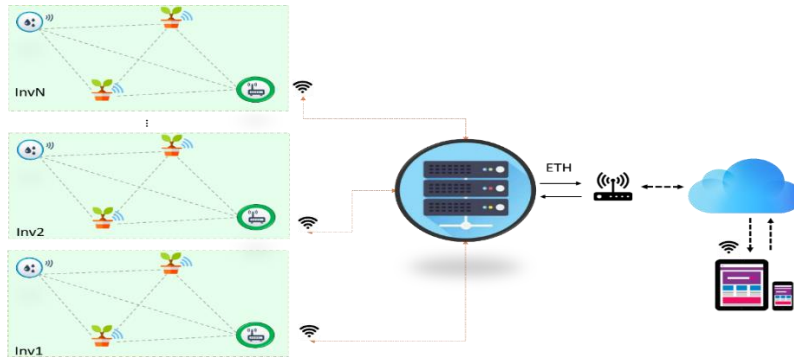


Fig. 9. Arquitectura general de red propuesta para el sistema de gestión inteligente para cultivos hidropónicos, caso múltiples invernaderos.

2.3. Capa de procesamiento y de aplicación

En esta capa se definieron los algoritmos de tratamiento de los datos recabados por los nodos Sensor y de Riego, mientras que en la capa de aplicación se realiza la implementación y presentación de la capa de procesamiento de datos para proveer los servicios evolucionados.

Dicho procesamiento es realizado por el Servidor (fig. 10), el cual, almacena los datos y de acuerdo con las acciones definidas, realiza la toma de decisiones de activación y desactivación de los actuadores por medio del cambio de estado del circuito de relevadores, para mantener el cultivo dentro de los valores óptimos para su desarrollo.

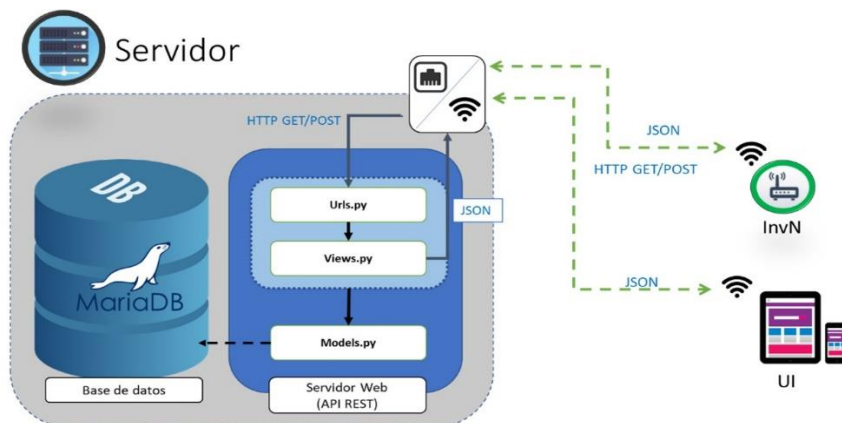


Fig. 10. Diagrama a bloques de la funcionalidad del Servidor del sistema de gestión inteligente.

2.4. Protocolo de comunicación Pouani

Durante el diseño del sistema de gestión basado en IoT para cultivos hidropónicos, se definió el protocolo de comunicación a nivel de aplicación “PouaniPonos”, para la

definición de la trama de comunicación de los valores obtenidos durante la fase de percepción, así como de los mensajes de configuración de los nodos. Más adelante profundizaremos en el diseño de dicho protocolo.

Preprocesamiento de los datos

Para definir la capacidad de memoria de la trama, es necesario considerar los valores mínimos y máximos que podría tener un parámetro en los diferentes casos de configuración y la precisión del dato. Por lo cual, se propuso realizar un preprocesamiento de los datos en cada nodo, con la finalidad de optimizar el espacio de memoria asignado a cada valor del parámetro obtenido en la capa de percepción.

Formato de trama

Una vez pre-procesados los valores, es necesario transmitirlos al servidor para realizar su procesamiento, almacenamiento y proporcionar los servicios. En la Fig. 11 se muestra el formato de la carga útil del nodo sensor y nodo riego en el caso de transmisión de las lecturas obtenidas de acuerdo con el formato de trama para la pila de protocolos BLE. Podemos ver que, con el diseño mínimo de la trama, se requiere 9 bytes de los 32 bytes disponibles por el transceptor nRF24L01+, por lo tanto, contamos con la capacidad de aumentar en un futuro el número de parámetros asignados a la fase de percepción dependiendo las características del proyecto.

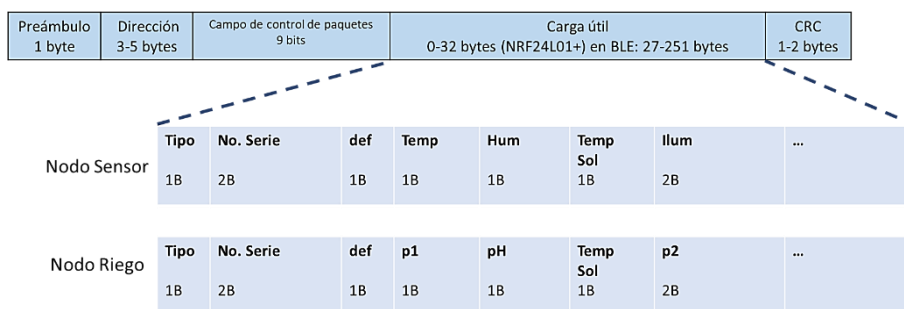


Fig. 11. Diagrama a bloques de la funcionalidad del Servidor del sistema de gestión inteligente

Transmisión del paquete – Creación de la red mallada Intra-Invernadero

Como se mencionó previamente, se establece el uso de una topología de red mallada dentro del invernadero como la adecuada para cubrir las necesidades de transmisión en dicha etapa, en específico, una red BLE mallada. Por lo cual, los nodos sensores, de riego y maestro deben ser configurados de tal forma que creen dicha red y transmitan entre sí los paquetes. El protocolo definido como “PouaniPonos” señala el procedimiento de tratamiento de la trama y establecimiento de la comunicación para el caso de sistemas de gestión agrícolas (Fig. 12).

Posteriormente se configuran en los segmentos de la memoria EEPROM los valores predeterminados de los rangos que funcionan como valores extremos para el monitoreo de las variables del entorno (temperatura, humedad, iluminación, pH, temperatura de la solución), la información de identificación del nodo (tipo de nodo y número de serie), configuración de los parámetros de control del sistema (periodos de muestreo, horarios en los que la configuración es válida y estado del nodo).

Después, se inicia la configuración y gestión de la red mallada por el nodo maestro. El cual, asigna de forma dinámica un identificador del nodo a los nodos esclavos (nodo sensor y nodo riego), promoviendo el registro dinámico de ellos, permitiendo el ingreso y egreso de nodos a la red (movilidad de los nodos).

La conexión es actualizada periódicamente con la finalidad de mantener el enlace entre los nodos y cumplir con los siguientes pasos:

1. Establecer canales de comunicación en donde el nodo es receptor de mensajes de configuración provenientes del nodo maestro o retransmisor de mensajes de lecturas obtenidas de otros nodos sensores o nodos riego (mensajes recibidos casos S y R).
2. Si recibe mensaje del nodo maestro (mensaje caso C), los valores son almacenados en el segmento de la memoria EEPROM del microcontrolador dedicada a valor de configuración definidas por el usuario.
3. Realizar el muestreo de los valores del entorno a través de los sensores de acuerdo con el periodo de muestreo configurado.
4. Procesar los valores obtenidos de acuerdo con los rangos definidos previamente.
5. Crear la trama con la carga útil en el formato diseñado para el mensaje con destino al nodo maestro.
6. Establecer el canal de comunicación al nodo más cercano y transmitir el mensaje.
7. Regresar al paso 1.

Protocolo "PouaniPonos"

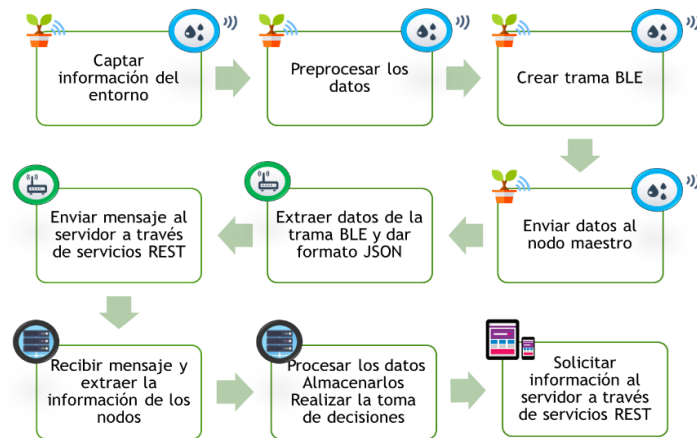


Fig. 12. Diagrama a bloques del Protocolo PouaniPonos.

Extracción de datos

Tras recibir los mensajes provenientes de los nodos sensor y de riego, el nodo maestro extrae los datos contenidos en el campo de carga útil de la trama BLE y los convierte a un objeto en formato JSON, para posteriormente transmitirlos al servidor mediante el uso de los servicios REST.

Transmisión del paquete – Red de comunicación Extra-Invernadero

Posterior a la extracción de los datos y su serialización al formato JSON, el nodo maestro requiere enviar los paquetes de mensajes al Servidor, en el cual, se establece

una comunicación en topología estrella, la cual permite la conexión de múltiples nodos maestros al servidor. Dicha red local es proveída por el servidor, el cual es configurado como punto de acceso inalámbrico (WAP) por medio del transceptor de tecnología WiFi incorporado.

La transmisión de los paquetes se realiza a través del consumo de servicios REST con la finalidad de agregar, eliminar, consultar y configurar los valores de configuración, lecturas e información de usuario a las rutas de los recursos asignados del servidor para proveer los servicios de almacenamiento y procesamiento de los datos.

Servicio de almacenamiento del servidor

Tras recibir las peticiones de servicios REST a los URLs definidos con los datos de las lecturas proveídos por los nodos, se realiza el procesamiento de dichos valores para visualizar la lectura del nodo de acuerdo con lo sensado (es decir, previo de ser pre-procesada). Posteriormente los datos son almacenados en las tablas de la base de datos del servidor de acuerdo con los modelos diseñados de la aplicación REST.

Despliegue de datos en la interfaz de usuario

Después de que el usuario realiza el inicio de sesión en la aplicación móvil, la vista inicial cumple con la función de desplegar la información de monitoreo de los cultivos, la figura 13 muestra el diagrama de vistas de la interfaz gráfica, en el cual, posterior a que una vista es llamada, se realiza el consumo de servicios REST para la consulta de los datos almacenados en la base de datos, estos son precargados y mostrados asincrónicamente en la interfaz de usuario.

Configuración de datos en la interfaz de usuario

La interfaz de usuario permite peticiones de agregar, configurar y eliminar información en el servidor a través del consumo de servicios REST. Dichas peticiones se realizan a través de las URLs asignadas que a su vez dirigen a los controladores que contienen el procedimiento adecuado de tratamiento a las peticiones. Para posteriormente actualizar los datos contenidos en las tablas de la base de datos.

Extracción y envío de los datos de configuración en el Servidor

Existe un segundo tipo de mensaje que implica a los nodos dentro del invernadero, este es el mensaje de configuración asignado a través de la interfaz de usuario. Para obtenerlo, el nodo maestro realiza peticiones de consulta al servidor a través de los servicios REST, en los cuales el servidor responde con un mensaje de configuración en formato JSON, que contiene los valores de previamente preprocesados con la finalidad de reducir el tamaño de memoria asignada a cada parámetro. Extrae los datos contenidos en el mensaje en formato JSON de-serializándolo y asignando los valores a una estructura de datos con propiedades, sin métodos.

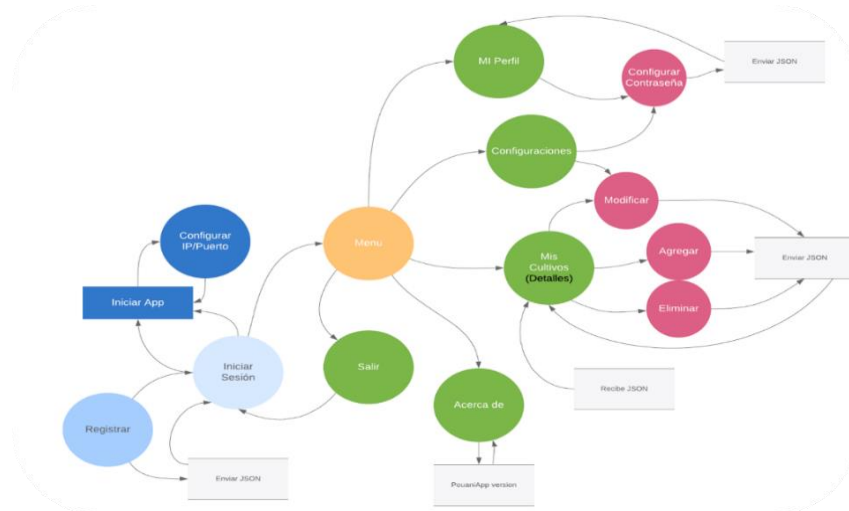


Fig. 13. Diagrama de vistas diseñadas para la interfaz gráfica de dispositivos móviles

Dichos valores se asignan a la carga útil de la trama BLE (Fig. 14) para su posterior transmisión a todos los nodos dentro del invernadero que contiene los valores máximos y mínimos de los parámetros, su estado (si están habilitados), periodo en que se tomará la muestra de sensado, hora de inicio y de fin en que la configuración es válida y en el caso del nodo riego, información referente a la actividad de riego (periodos de riego y duración), en donde se transmitirán 21 bytes de datos.

Previamente, se define en el campo de encabezado de la trama BLE el tipo de mensaje, siendo tipo “S”, el mensaje de configuración para el nodo sensor y el tipo “R” el mensaje de configuración del nodo riego. Por lo que, al recibir el mensaje, los nodos leen el encabezado de la trama, específicamente el tipo de mensaje y con ello filtran el tipo de nodo destinatario de dicha configuración y después se realiza la actualización de valores de configuración en la memoria EEPROM de cada nodo.

Trama BLE

Preámbulo	Dirección	Campo de control de paquetes	Carga útil	CRC
1 byte	3-5 bytes	9 bits	0-32 bytes (NRF24L01+) en BLE: 27-251 bytes	1-2 bytes

Tipo	def	Temp Max	Temp Min	Hum Max	Hum Min	TempSol Max	TempSol Min	IlumMax	IlumMin	Periodo Muestreo	Hora inicio	Min inicio	Hora fin	min fin
1 byte	bool	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes	1 byte	1 byte	1 byte	1 byte

Nodo Sensor

Tipo	def	P1 Max	P1 Min	pH Max	pH Min	TempSol Max	TempSol Min	P2 Max	P2 Min	Periodo Muestreo	Hora inicio	Min inicio	Hora fin	min fin
1 byte	bool	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes	1 byte	1 byte	1 byte	1 byte

Nodo Riego

Fig. 14. Estructura de datos del mensaje de configuración de los nodos sensor y de Riego en formato BLE.

2.5. Configuración y comunicación con el servidor

El servicio de base de datos para contar con un historial del estado del entorno de los cultivos (obtenidos por los nodos sensores y de riego) y el servicio web de aplicaciones REST para proveer los modelos, controladores y urls necesarios. Como se mencionó en los capítulos anteriores, la base de datos fue diseñada sobre el gestor MariaDB que permite crear modelos relacionales entre los campos definidos para la solución, mientras que los servicios REST fueron configurados sobre Django, permitiéndonos crear modelos simples y controladores para proveer las acciones diseñadas a través de diversas peticiones por medio del consumo de los servicios REST a través de una interfaz de usuario o desde los nodos maestro.

2.6. Creación de la interfaz de usuario

La interfaz de usuario permite la interpretación del estado del cultivo a través de una aplicación, mostrando sus variaciones a lo largo de la campaña de cultivo. El usuario puede apoyarse en ella para la toma de decisiones respecto a la activación y desactivación de los actuadores para regular dichas variables. Con ello, el riesgo de daño o abandono durante su desarrollo o periodos críticos puede disminuir y potencializa la probabilidad de obtener productos de mayor calidad y uniformidad.

Las Figs. 15 y 16 muestran las vistas de la interfaz de usuario creada para terminales móviles bajo Sistemas Operativos Android. En las cuales se cubren las necesidades de autenticación básica a través de una validación de usuario / contraseña, el despliegue de la información del estado del cultivo a través de las lecturas de los nodos y los servicios de agregar, remover, modificar la información de los cultivos y la asociación de la información de ubicación del cultivo y nodos.

De igual forma se configuraron validaciones de llenado de las formas para asegurar que el usuario llene los campos, previo a solicitar las peticiones al servidor. Así como una vista para realizar las configuraciones de la información para conectividad con el servidor de acuerdo con su dirección IP y puerto.

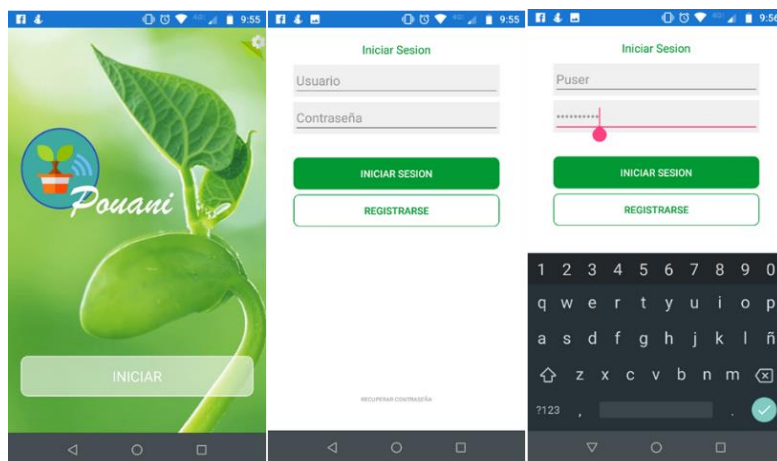


Fig. 15. Vista de Error al intentar Iniciar sesión sin usuario o contraseña (derecha) y vista del formulario de Registro de usuario

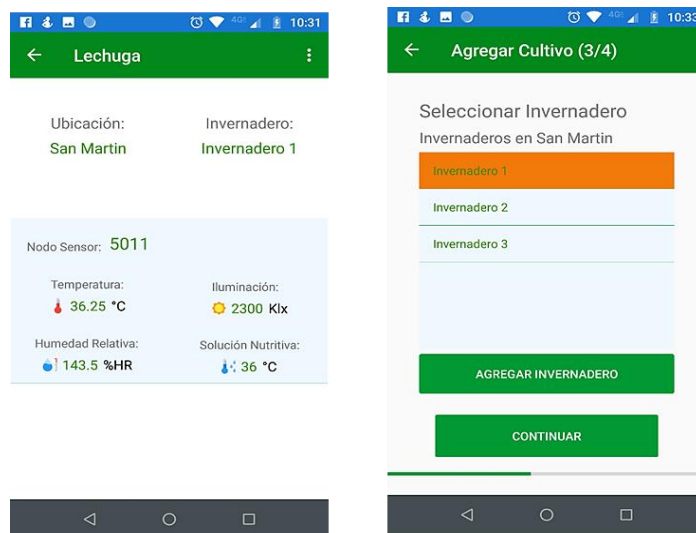


Fig. 16. Vistas de Configuración y Monitoreo.

3. Conclusiones

Debido al rápido crecimiento de la población y la alta demanda que esto produce, es necesario desarrollar sistemas que apoyen las necesidades básicas de los pueblos, por ello, los sistemas de agricultura inteligente han destacado ampliamente por sus aportaciones en el sector agrícola en diversos países, en donde las condiciones del entorno son poco favorables, pero el acceso a la tecnología y su implementación en los cultivos permite reducir la hostilidad y los riesgos durante el desarrollo y crecimiento de las plantas, permitiendo que los métodos agrícolas evolucionen de conocimientos empíricos a ser métodos profesionales que puedan ser medidos cualitativa y cuantitativamente.

El principal enfoque del presente proyecto es crear equipos que provean las herramientas al sector agrícola, que sean de fácil acceso y uso, con la finalidad de reducir la brecha tecnológica existente entre los productores agrícolas de los sectores mini, microempresa y de autoconsumo respecto a los productores intensivos, los cuales cuentan con una mayor capacidad adquisitiva para implementar soluciones de gestión y automatización en sus cultivos.

Con lo cual el diseño de una solución aplicable a la agricultura hidropónica requirió de múltiples esfuerzos para cubrir los principales requerimientos para dichos sistemas. El diseño modular de la solución nos permitió desarrollar equipos capaces de crecer en el sentido de la capacidad de adquisición de los datos. Esto es, a nivel de prototipo, los nodos pueden obtener la información de los parámetros clave del entorno definidos en el diseño, pero de acuerdo con las necesidades de monitoreo del agricultor, se dispone de memoria y de puertos para agregar o modificar los sensores y actuadores, permitiendo con ello crear diversos productos de monitoreo específicos a partir del diseño base de los equipos.

La topología de red del sistema fue basada en un diseño modular y escalable, que permita el crecimiento del sistema de acuerdo con las necesidades del productor. Por lo que podemos definir el producto en un paquete de inicio que contenga los equipos mínimos para modernizar el cultivo. Este paquete inicial contiene el nodo sensor, nodo de riego, nodo maestro, servidor y aplicación móvil, los cuales proveen la gestión del cultivo para un área de cobertura de un módulo hidropónico. Posteriormente y de acuerdo con el crecimiento del sistema de cultivo o la capacidad adquisitiva del productor, éste puede obtener nodos sensores o de riego adicionales para ampliar la cobertura de monitoreo y gestión del sistema e incorporarlos sin necesidad de reconfigurar la red, siendo un proceso transparente al usuario.

Además, es claro que la tecnología avanza y con ellos los protocolos de comunicación evolucionan, por ello se diseñó un modelo flexible el cual permita, en un futuro, modificar los protocolos de comunicación a tecnologías más recientes con afectaciones mínimas para el usuario.

La definición del protocolo de formato y tratamiento de la trama a nivel aplicación “PouaniPonos” pretende normalizar los métodos de comunicación y tratamiento de los datos para sistemas enfocados a la agricultura inteligente. Las ventajas del protocolo de trama “PouaniPonos” permite la optimización del espacio de memoria asignado a la carga útil de la trama a través del preprocesamiento de los datos recabados y con ello aprovechar dicho espacio de memoria para la transmisión de múltiples parámetros por nodo. Así mismo, el diseño de la topología de red híbrida que aprovecha las fortalezas de las tecnologías de comunicación WiFi y Bluetooth para entornos agrícolas, demostró ser viable en condiciones de laboratorio. De acuerdo con la tendencia de comportamiento y respuesta en términos de funcionalidad, el siguiente paso en el desarrollo del sistema, es implementar la solución y pruebas en campo, con la finalidad de complementar el análisis y resultado del comportamiento de los equipos en condiciones reales.

Finalmente, se plantea un modelo general de cultivos hidropónicos. El cual, se espera presente una base para futuros sistemas para agricultura inteligente, basados en tecnologías de nueva generación, los cuales eventualmente permitan a la mayoría de la población, contar con una fuente de alimento sustentable, eficiente y de mejor calidad mediante el uso de sistemas de cultivos hidropónicos personales.

Referencias

1. Servicio de Información Agroalimentaria y Pesquera (SIAP): Uso de tecnología y servicios en el campo Cuadros tabulares (2016)
2. World Bank, CIAT, CATIE: Climate-smart agriculture in Mexico. CSA country profiles for Latin America Series. World Bank Gr (2014)
3. Oasis, S.: Manual de hidroponia. 32 (2002)
4. Patokar, A.M., Gohokar, V.V.: Precision Agriculture System Design Using Wireless Sensor Network. pp. 169–177, doi: 10.1007/978-981-10-5508-9_16 (2017)
5. Verdouw, C.: Internet of Things in agriculture. CAB Rev Perspect Agric Vet Sci Nutr Nat Resour 11: doi: 10.1079/PAVSNNR201611035(2016)
6. Kiani, F., Seyyedabbasi, A. Wireless Sensor Network and Internet of Things in Precision Agriculture. Int J Adv Comput Sci Appl 9, 99–103 doi: 10.14569/ijacsa.2018.090614 (2018)
7. Azcón-Bieto, J., Talón, M.: Fundamentos de fisiología vegetal (2003)
8. Araceli, M.: Manual de hidroponia, 1a Ed. Instituto de Biología, Distrito Federal (2014)

9. Minerva, R., Biru, A., Rotondi, D. Towards a definition of the Internet of Things (IoT). IEEE Internet Things 86. doi: 10.5120/19787-1571 (2015)

Propuesta de sistema de gestión para optimización de redes de transporte público

Jorge Erik Montiel Arguijo¹, Evelyn Gutiérrez León¹, Chadwick Carreto Arellano², Felipe Rolando Menchaca García¹

¹ Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica unidad Zacatenco, Ciudad de México, México

² Instituto Politécnico Nacional, Escuela Superior de Cómputo, Ciudad de México, México
jerikmontiel.isc@gmail.com, evelyngleon@hotmail.com
ccarreto@ipn.mx, fmenchac@gmail.com

Resumen. El presente trabajo propone el desarrollo de un sistema de monitoreo y administración inteligente de redes de transporte público utilizando una arquitectura de comunicación híbrida basada en redes de nodos inalámbricos que utilizan IPv6 y red celular (LTE, LTE-M), está destinado a describa los diferentes elementos necesarios para la recopilación de información, comunicación, procesamiento, almacenamiento y posterior despliegue de información.

Palabras clave: ciudades inteligentes, redes inalámbricas de sensores, internet de las cosas, nodos inalámbricos, arquitectura de comunicación.

Proposal of a Management System for Optimization of Public Transport Networks

Abstract. The present work proposes the development of a system of intelligent monitoring and administration of public transport networks using a hybrid communication architecture based on networks of wireless nodes using IPv6 and cellular network (LTE, LTE-M), it is intended to describe the different elements necessary for the collection of information, communication, processing, storage and subsequent deployment of information.

Keywords: smart cities, wireless sensor networks, internet of things, wireless nodes, communication architecture.

1. Introducción

Conforme las ciudades siguen creciendo en tamaño y población, el diseño de redes de transporte público se vuelve complicado, dada la amplia diversidad en los orígenes y destinos de los usuarios [1]. Así como la saturación de la infraestructura vehicular en las grandes ciudades a pesar de sus intentos de crecerla de acuerdo con la distribución

de la población. Lo cual denota que, para reducir el tiempo de traslado de los usuarios es necesario implementar soluciones viales alternativas al uso del auto-móvil, aumentando la inversión en el transporte público [2 y 3] realizando un análisis integral del estado del transporte.

Esto ha generado que, actualmente a nivel global, se busquen soluciones y desarrollo orientado a transporte basadas en Internet de las cosas (IoT) que permitan, en una primera etapa, monitorear los sistemas de transporte público, con el fin de optimizar el despliegue de las unidades de transporte y con ello reducir el tiempo de traslado de los usuarios a través de las ciudades [4 a 6]. Dichas propuestas de solución son enfocadas a información recolectada de los recursos de los usuarios (datos recolectados por medio de los teléfonos inteligentes) para crear una base de datos común [3].

La Ciudad de México (CDMX) no se exime de estos problemas, dado que, su población (incluyendo los municipios conurbados del Estado de México), ha crecido rápidamente en los últimos años, lo que ha provocado que el tiempo de traslado entre los hogares y oficinas de los habitantes de la ciudad sea mayor, siendo ubicado entre las categorías de contendiente y con un nivel de aspiración importante según el índice de movilidad urbana publicado por Deloitte [7].

En el presente artículo se presenta la primera parte de una plataforma escalable que permite conectar diversas fuentes de monitoreo e información mediante una red híbrida, conformada por diferentes redes inalámbricas.

Esta plataforma está diseñada para adaptarse a las nuevas necesidades de la ciudad conforme se implemente, ofrece una forma de intercomunicación con sistemas nuevos y existentes que permita iniciar la conversión a una ciudad inteligente, atacando, primeramente, uno de los problemas más importantes en este momento: El transporte público.

La propuesta se centra en resolver en primera instancia, el reto de conocer el estado del transporte público, dividiéndolo en tres partes esenciales: En la primera parte se abordan los módulos generales de hardware y software necesarios para el correcto funcionamiento del sistema, posteriormente se describe brevemente la arquitectura de red que se utilizará y las tecnologías y protocolos necesarios para la operación de la red, finalmente en la tercera parte se describe la arquitectura general del sistema y se define la forma de comunicar la información recopilada desde la red de autobuses a un sistema de información para su posterior despliegue a los dispositivos de los usuarios.

Más adelante, se describe brevemente el análisis existente hasta el momento para el desarrollo del sistema de información y las conclusiones obtenidas.

2. Desarrollo

Con base en las características de los diferentes tipos de sistemas de transporte analizados (Metrobús, Trolebús y RTP) y en las diferentes necesidades de los usuarios, se propone una solución modular, escalable y mínimamente invasiva, la cual, utiliza tecnologías de nueva generación y bajo costo. Estas características permiten que sea posible la implementación de los dispositivos de monitoreo y comunicación de forma sencilla, haciendo uso de infraestructura existente. A continuación, se describen de a

manera general los elementos que conforman el sistema a nivel de hardware y a nivel de software.

2.1. Elementos del sistema

Como se mencionó previamente, en la primera etapa, el sistema de gestión se comportará más como un sistema de información, en donde se recopilan datos de las diferentes unidades de transporte y las diferentes estaciones, a partir de sensores distribuidos en estaciones y unidades, así como de datos recopilados desde los dispositivos móviles de los usuarios. Es importante recalcar que este sistema busca concentrar y comunicar información de otros sistemas que se encuentran actualmente en desarrollo y no busca sustituirlos, es decir, en primera instancia y a diferencia de otros sistemas, no se recopila la totalidad de la información desde los dispositivos de los usuarios, en cambio se obtiene de la red de sensores en las unidades para su posterior despliegue al usuario mediante una plataforma móvil.

Los elementos del sistema se dividen en: Módulos de monitoreo, que incluye los módulos de autobús y módulos de estación; Aplicación web y móvil y Sistema de información.

Monitoreo de autobús (A: Thread y B: Thread+LTE)

Con el objetivo de informar del nivel de saturación de las líneas de transporte, se implementarán módulos que capaces de obtener el número de usuarios en la unidad en todo momento, de igual forma para conocer la ubicación de las unidades se instalará un módulo GPS el cual operará en conjunto con un transceptor BLE para detectar dispositivos de microlocalización “Beacons” para informar al usuario acerca de la estación en la que se encuentra por medio de una aplicación web del sistema. Como se puede ver en la fig. 1 se propone distribuir este monitoreo en dos tipos de módulos de acuerdo con el tipo de transporte en el que se implementará el sistema. Para sistemas de transporte con estaciones señalizadas, por ejemplo: RTP, se utilizarán nodos de tipo B, en cambio, para sistemas de transporte con estaciones con estructuras fija, ejemplo MB, se pueden utilizar nodos de tipo A o B de manera indistinta.

El diseño de los nodos es conformado de la siguiente manera:

- Unidad de control.
- Contadores de ascenso/descenso con detección de dirección.
- Módulo GPS
- Módulo de lectura de Beacons
- Módulo de comunicación Thread
- Módulo de comunicación LTE (Solo tipo B)

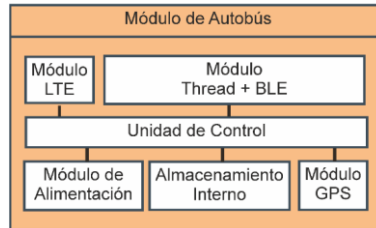


Fig. 1. Diagrama del módulo central del autobús.

Con base en la gran diversidad de tipos de unidades de transporte, se definieron diferentes módulos de monitoreo de ascenso/descenso, comunicados inalámbricamente con un módulo central mediante una red Bluetooth Mesh, otorgando una gran flexibilidad para instalar los sensores necesarios en cada unidad. La fig. 2 muestra el diagrama a bloques de los módulos de ascenso/descenso.

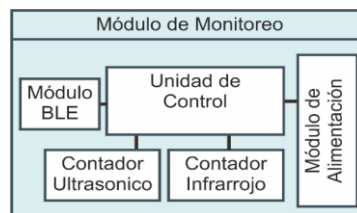


Fig. 2. Diagrama del módulo de monitoreo ascenso/descenso del autobús.

Los módulos de monitoreo se distribuirán en cada uno de los puntos de ascenso y descenso de la unidad, siendo los encargados de obtener la información de ascensos y descensos en cada punto de la unidad, y comunicarlos al módulo central de autobús para calcular el número de personas actual dentro de la unidad.

Módulos de estación

Como se mencionó previamente, este sistema se enfoca en distribuir los dispositivos en los autobuses, con el objetivo de garantizar versatilidad de la solución, en otras palabras, que todo sistema de transporte sea aplicable a la solución con los menores ajustes posibles. Sin embargo, es necesario para una correcta sincronización y despliegue de información al usuario, colocar módulos de identificación en las estaciones, estos módulos pueden ser completamente autónomos y con mínimo impacto en la infraestructura actual. De acuerdo con el tipo de sistema de transporte en el que se implementará la solución, se proponen dos tipos de módulos:

Estaciones señalizadas

En el caso de las rutas que sólo dispongan de estaciones señalizadas, se colocaran “Beacons” en cada estación, este dispositivo permitirá, con base en la intensidad de señal, obtener la proximidad de los autobuses respecto a la estación, esto es, si está llegando, saliendo o si está parado en la estación. Este en conjunto con el módulo GPS, se obtendrá información de la ubicación de la unidad; además este dispositivo permite

a los usuarios que cuenten con la aplicación móvil en sus dispositivos, visualizar en qué estación se encuentran de forma automática y sin consumir de su paquete de datos o su módulo GPS, siempre y cuando sean compatibles con el protocolo a utilizar en los Beacons.

Estaciones de estructura fija

Para las estaciones con estructura fija como lo son estaciones de sistemas BRT (Metrobús, Mexibús, Algunos trolebuses, y autobuses del sistema M1) se colocará adicionalmente al Beacon un módulo de comunicación LTE, esto para brindar puntos de redundancia adicional en la red y con ello garantizar que el sistema de información siempre estará actualizado incluso sin autobuses con nodos tipo B disponibles “Fig. 3”.

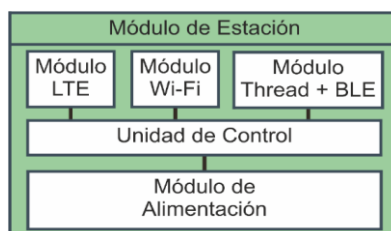


Fig. 3. Diagrama de módulo de estación con estructura fija.

Estos módulos solamente actúan como repetidores y nodos frontera de la red Thread desplegada entre las estaciones y los autobuses, la característica principal a resaltar en estos nodos es la posibilidad de adaptarlo a la tecnología de comunicación a nivel de internet que se tenga disponible en el lugar de instalación, es decir, se puede implementar la comunicación del sistema a nivel de internet a partir de un módulo, LTE, 3G, Wifi o Ethernet, entre otros.

Aplicación Web

Para el despliegue de información a los usuarios se desarrollará una aplicación web la cual permita acceder a la información del sistema desde cualquier dispositivo compatible, de igual forma se adaptará esta aplicación web para que se ejecute como una aplicación en teléfonos inteligentes, primeramente, con sistema operativo Android y posteriormente iOS.

La aplicación móvil permitiría a los usuarios del transporte visualizar el estado de las líneas incluyendo: tiempo de llegada del próximo autobús a una parada específica, nivel de saturación del autobús (cantidad de personas/capacidad específica), diferentes rutas de transporte registradas en el sistema que paran en una misma parada, así como detectar automáticamente la parada en la que se encuentra mediante la detección del “Beacon” instalado en la parada.

En el caso de los sistemas de transporte BRTs se puede desplegar la información de los autobuses, así como su tiempo estimado de llegada en los monitores instalados dentro de las estaciones.

A los operadores y administradores de la red de transporte, la aplicación les permitirá visualizar indicadores de uso de la red incluyendo ascensos y descensos por parada,

nivel de saturaciones de los autobuses por estación, ruta y hora, así como el tiempo entre autobuses.

Así mismo se busca que la página permita a los administradores de la red, generar indicadores de operación los cuales les permitan una mejor toma de decisiones con base en el uso de la información recopilada por el sistema en “tiempo real”.

2.2. Sistema de información

La información recabada en los sistemas de transporte se comunicará mediante los módulos LTE con el sistema de información. Este es el encargado de procesar y almacenar toda la información de la red de forma que esta pueda ser desplegada de forma fácil y entendible mediante gráficos e indicadores de las diferentes variables del sistema.

El sistema de información se alojará en una nube privada y se gestionará mediante el uso de servicios REST. Los cuales interconectaran los diferentes nodos del sistema, así como los clientes móviles, y aplicaciones web.

2.3. Arquitectura de red

Para la comunicación de los diferentes puntos de monitoreo se propone una arquitectura de red híbrida conformada por una red de nodos inalámbrica la cual brinde una primera infraestructura de comunicación entre las unidades de transporte y un conjunto de enlaces LTE que permitan comunicar todos los nodos a un sistema de información en la nube el cual distribuya la información a los usuarios de la red de transporte (Fig. 4).

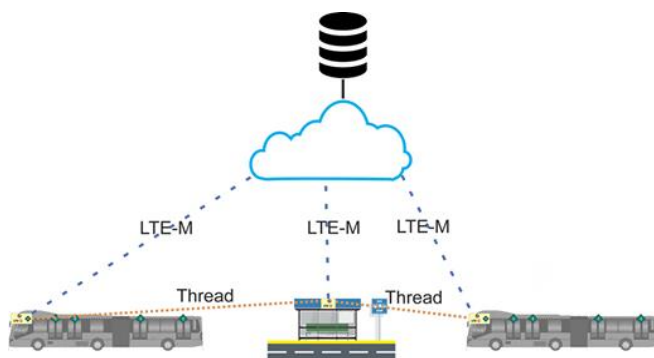


Fig. 4. Arquitectura de comunicación básica.

Dentro de las unidades de transporte es necesario colocar nodos de monitoreo en diferentes puntos de la unidad, estos puntos se comunicarán inalámbricamente mediante una red Bluetooth Mesh. La información se concentrará en el nodo principal del vehículo el cual comunicará las variables monitoreadas con los nodos cercanos utilizando la red Thread y en caso que el vehículo disponga de un modem LTE se enviará la información igualmente por este medio al sistema de información “fig. 5”.

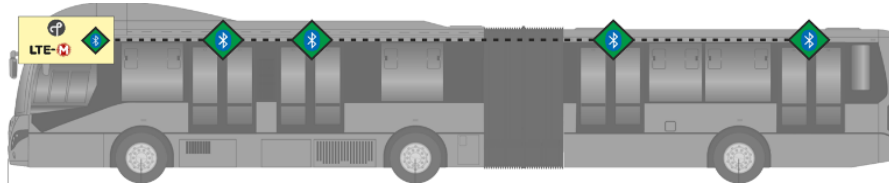


Fig. 5. Distribución de nodos en unidades de transporte.

El uso de una topología en malla permite agregar y configurar los elementos necesarios para el monitoreo de forma dinámica con base en las características del transporte específico a monitorear.

Para interconectar las diferentes unidades de transporte con las estaciones de estructura fija y posteriormente comunicarse con el sistema de información, en las estaciones se desplegarán nodos basados en Thread los cuales permitirán una comunicación directa entre estaciones y unidades de transporte optimizando la distribución de las variables monitoreadas en cada punto de forma independiente ayudando con esto a alimentar los algoritmos de aprendizaje automático que se implementarán en la red de nodos “Fig. 6”, de igual forma que en las unidades de transporte se seleccionarán algunas estaciones específicas (con base en su ubicación geográfica) para incluir en su nodo un modem LTE que permita comunicar los nodos con el sistema de información.

Es importante mencionar que la red de nodos también obtendrá variables de configuración y de control del sistema de información las cuales permitirán ajustar los parámetros de los algoritmos de aprendizaje automático con base en las necesidades de los usuarios (gestores de la red de transporte).

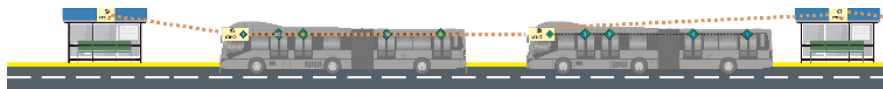


Fig. 6. Comunicación nodal entre estaciones y autobuses.

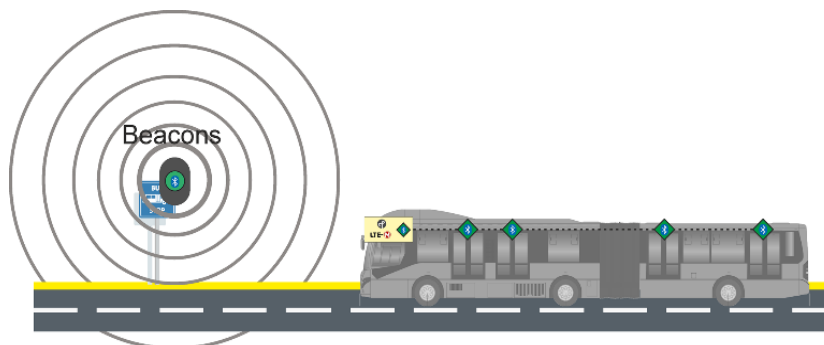


Fig. 7. Detección por proximidad en estaciones señalizadas.

En el caso de las estaciones que no dispongan de las características necesarias para operar un nodo de comunicación, como los son aquellas conformadas solo por un letrero en una vialidad, se colocarán dispositivos de microlocalización "Beacons" los cuales permitan a las unidades de transporte conocer y posteriormente reportar la ubicación de la unidad mediante el identificador de la estación "Fig. 7"; y en el caso de los usuarios con dispositivos móviles compatibles obtener información de las próximas unidades de transporte mediante la aplicación móvil desarrollada.

Finalmente, los usuarios del sistema (pasajeros, operadores y gestores de la red de transporte) visualizarán la información recopilada del sistema mediante una aplicación web en cualquier dispositivo compatible con conexión a internet.

2.4. Arquitectura del sistema

Para la comunicación a nivel lógico de todos los componentes del sistema se propone una arquitectura de software modular la cual permite el monitoreo y la notificación a los módulos mediante el uso de servicios REST para el primer caso y notificaciones PUSH para el segundo.

Esta arquitectura se simplifica en gran parte en la comunicación con los nodos Thread debido a que este protocolo permite re direccionar las tramas de información de forma directa gracias a la implementación de IPv6 dentro de la comunicación de los nodos [8].

Los nodos distribuidos en los autobuses comunican la información recabada a través de la red Bluetooth Mesh desplegada dentro del autobús, posteriormente esta se comunica mediante la red Thread hasta alcanzar un nodo frontera a partir del cual se comunicará al servidor Web haciendo uso de los servicios Web.

Una vez en el servidor esta información se procesará y se almacenará en la base de datos del sistema de información y posteriormente, con base en la fuente de origen de la información se notificará a los nodos y usuarios en las estaciones pertinentes mediante los canales de notificaciones PUSH con lo cual se evitará que la aplicación Web y los nodos de la red estén realizando un "poleo" constante.

3. Implementación

Hasta este momento se ha llevado a cabo el análisis y diseño de los módulos de monitoreo dentro del autobús los cuales manejan la cuenta de usuarios dentro de las unidades de transporte, se desarrollaron 2 prototipos de este modo, el primero utilizando el transceptor nRF24L01+ con el objetivo de comunicar los diferentes nodos instalados dentro de una unidad. Debido a la modularidad del sistema se pueden modificar las características de la red de sensores inalámbrica, por ello el segundo prototipo se optó por utilizar un conjunto de sensores y transceptores diferentes, en este segundo caso, se agregó un sensor de corte infrarrojo en conjunto con los sensores de distancia infrarrojos y un XBee S2C como módulo de comunicación entre nodos.

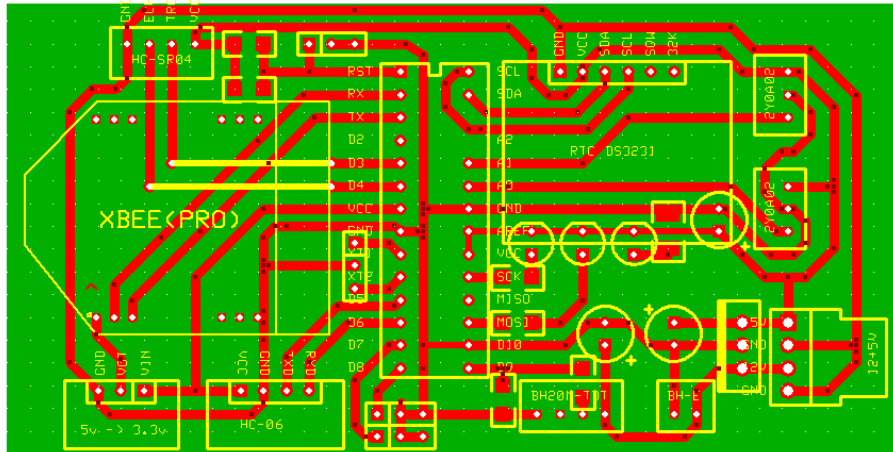


Fig. 8. Diseño de PCB de módulo de monitoreo usando XBee

Finalmente se diseñó un módulo adaptable al cual puede conectarse cualquier de los dos transceptores, así como los diferentes sensores analizados, este último diseño se realizó en EAGLE y busca ser compacto y compatible cualquiera de los transceptores probados, utilizando ZigBee a través de XBee mediante el módulo XB1 o mediante BLE a través del nRF24L01+ usando el conector JP3.

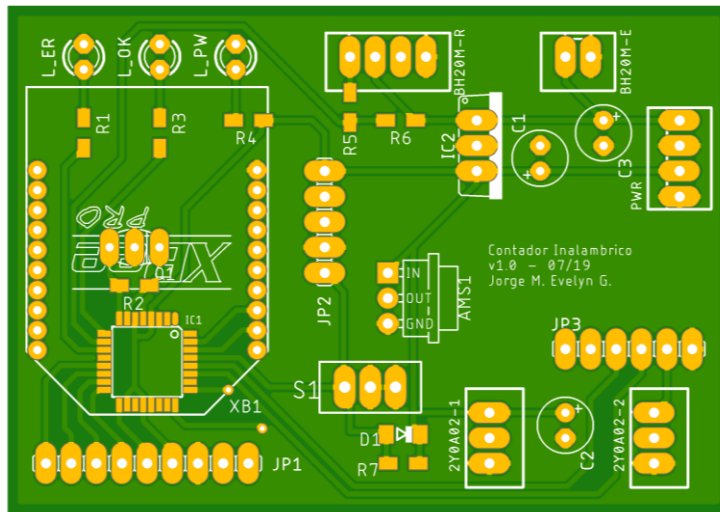


Fig. 9. Diseño de PCB de nodo de monitoreo en EAGLE

De igual forma se trabaja con tarjetas de desarrollo basadas en el nRF52840 para el despliegue de los nodos de autobús y de estación, estos módulos son los encargados de generar la red Thread en multiprotocolo con BLE para la comunicación de las diferentes unidades de transporte.

4. Conclusiones

El sistema propuesto se distingue de otras implementaciones similares en la arquitectura de comunicación de los vehículos de transporte la cual le permite al nodo conocer el estado de los demás nodos además del suyo.

La propuesta tiene como principales ventajas:

- Arquitectura fácilmente escalable.
- Nodos inalámbricos que permiten alta flexibilidad en la instalación dentro de las unidades de transporte.
- Comunicación de las unidades de transporte sin necesidad de conexión a internet.
- Posibilidad de utilizar diversos tipos de unidades de transporte y diversas rutas dentro de un mismo sistema.

Como trabajo a futuro se desarrollará la red de comunicación entre autobuses y estaciones basada en el protocolo Thread, se desarrollarán los módulos de comunicación LTE-M y se desarrollará el sistema de información con las tecnologías de comunicación necesarias para poder verificar el correcto funcionamiento del sistema, igualmente se desplegarán algunos nodos en vehículos designados los cuales se monitorearán en un camino definido dentro de la ciudad esto con el fin de realizar pruebas de operación en un entorno real y verificar la funcionalidad del sistema.

Se considera en una siguiente etapa visualizar la posibilidad de integrar el pago del servicio de transporte a través de la plataforma, así como fungir como credencial de identificación para el acceso a la ruta de transporte. Esto puede ser desarrollado mediante la implementación de NFC o a través de algún código bidimensional dinámicos (QR, PDF417, etc.).

Finalmente consideramos que es importante el desarrollo de soluciones enfocadas a IoT, entre ellas a Smart Cities que permitan la integración de la diversa infraestructura existente a las nuevas tendencias en sistemas de información, de forma que se puedan optimizar los diferentes servicios que ofrecen las ciudades, reduciendo, en el caso del transporte público, los tiempos de traslado y mejorando la eficiencia en el consumo de combustible, mejorando paulatinamente la calidad de vida de los habitantes.

Referencias

1. Handte, M., Foell, S., Wagner, S., Kortuem, G., Marron, P.J.: An Internet-of-Things Enabled Connected Navigation System for Urban Bus Riders. *IEEE Internet Things J.* 3, 735–744 (2016). <https://doi.org/10.1109/JIOT.2016.2554146>
2. Instituto de investigaciones parlamentarias. Diagnóstico de movilidad en la Ciudad en México: El impacto del crecimiento vehicular (problemas, estadísticas y evaluación de políticas), 118 (2017)
3. Farkas K, Nagy, A.Z., Tomas, T., Szabo, R.: Participatory sensing based real-time public transport information service. In: 2014 IEEE Int Conf Pervasive Comput Commun Work PERCOM Work 2014, pp. 141–144 (2014). doi: 10.1109/PerComW.2014.6815181

4. Lohokare, J., Dani, R., Sontakke, S., Adhao, R.: Scalable tracking system for public buses using IoT technologies. In: 2017 Int. Conf. Emerg. Trends Innov. ICT, ICEI 2017, pp. 104–109 <https://doi.org/10.1109/ETIICT.2017.7977019> (2017)
5. Raj, J.T., Sankar, J.: IoT based smart school bus monitoring and notification system. In: 5th IEEE Reg. 10 Humanit. Technol. Conf. 2017, R10-HTC 2017, pp. 89–92. <https://doi.org/10.1109/R10-HTC.2017.8288913> (2018)
6. Reyes, C., Barreto, A., Bautista, V.E.: Sistema de monitoreo del LOBOBUS. 2, 998 (2015)
7. Nieblas, M.: Movilidad en la CDMX | Deloitte México. <https://www2.deloitte.com/mx/es/pages/dnoticias/articulos/movilidad-en-la-cdmx.html>
8. Unwala, I., Taqvi, Z., Lu, J.: Thread: An IoT protocol. In: IEEE Green Technol. Conf, pp. 2018-April, 161–167. <https://doi.org/10.1109/GreenTech.2018.00037> (2018)

Prototipo de sistema embebido para la asistencia de personas con discapacidad visual en la acera pública

Diego Cruz Mondragón, Katia Santillán Álvarez, Víctor Hugo García Ortega,
Juan Humberto Sossa Azuela

Instituto Politécnico Nacional, Escuela Superior de Cómputo, Ciudad de México,
México
diegocruzmondragon@hotmail.com, kat_santillan@outlook.com

Resumen. Se plantea el desarrollo de un prototipo de sistema embebido para la asistencia en el desplazamiento de personas con discapacidad visual. Este sistema estará montado en un computador de placa reducida, el cual, sea capaz de tomar una fotografía del entorno en el que se encuentra el usuario del sistema, para posteriormente procesarla y detectar los posibles objetos que puedan interferir en su desplazamiento. Finalmente, mediante síntesis de voz, generar un comando de alerta que permita evitar los objetos identificados.

Palabras clave: asistencia a discapacitados, reconocimiento de formas, sistema embebido, discapacidad visual, electrónica digital.

Embedded System Prototype for the Assistance of Visually Impaired People on the Public Sidewalk

Abstract. The development of a prototype of the embedded system for the assistance in the displacement of people with visual disabilities is proposed. This system will be mounted on a reduced plate computer, which is capable of taking a picture of the environment in which the user of the system is located, to subsequently process it and detect possible objects that may interfere with its movement. Finally, by voice synthesis, generate an alert command that allows to avoid the identified objects.

Keywords: assistance for the disabled, recognition of forms, embedded system, visual impairment, digital electronics.

1. Introducción

De acuerdo con datos del INEGI, en 2014 el 6.4% de la población del país (7.65 millones de personas) reportaron tener al menos una discapacidad, de

este porcentaje, se estima que, alrededor del 35.5%, presentan algún tipo de discapacidad visual, total o parcial, lo que la convierte en la segunda causa de discapacidad visual, después de la motriz.

Las dificultades relacionadas con la visión representan retos para las personas que las padecen y están ligadas a la complejidad de enfrentarse a un entorno que no siempre se encuentra adaptado a sus necesidades, lo cual tiene como consecuencia, de acuerdo con la CIF, que la actividad y la participación de la persona disminuya [1].

2. Metodología

El desarrollo de soluciones basada en sistemas electrónicos embebidos de alta confiabilidad va en aumento y los requisitos de los entornos a los que se orienta el prototipo, son principalmente el coste, el tiempo y la confiabilidad, por lo tanto, se observa la necesidad de una metodología, como herramienta, que facilite el desarrollo de sistemas electrónicos embebidos de alta confiabilidad [2].

El prototipo generado abarca aspectos tanto de Software (SW) como de Hardware (HW), porque pretende dar solución a problemas reales donde resulta necesario la interacción de ambos.

Con base en lo anterior se decidió utilizar la metodología en V o de 4 niveles, ya que el realizar pruebas en cada una de las fases permitirá corregir los errores detectados con mayor antelación sin esperar hasta el termino de cada etapa para rectificar el correcto funcionamiento del sistema, asimismo, este modelo permite que las estimaciones de fechas y presupuesto para las pruebas unitarias y de integración sean determinadas con mayor precisión.

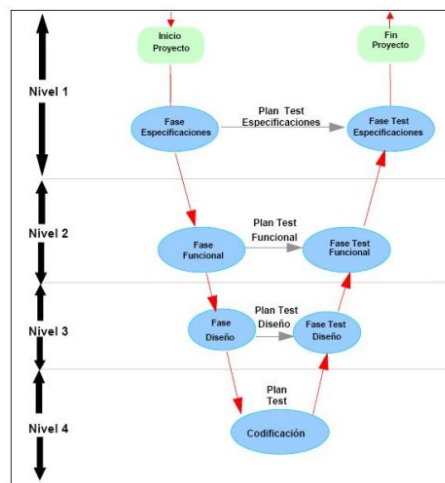


Fig. 1. El modelo representa, en forma de V, las relaciones temporales entre las distintas fases del ciclo de desarrollo de un proyecto.

En cada uno de los niveles mostrados en la Figura 1 existe una correspondencia, existe una fase correspondiente o paralela de verificación o validación. Esta estructura obedece al principio de que para cada fase del desarrollo debe existir un resultado verificable.

2.1. Sistemas embebidos

Los sistemas embebidos son dispositivos de propósito específico que han sido diseñados para realizar un conjunto reducido de operaciones.

2.2. Sensores

Son dispositivos que están capacitados para detectar acciones o estímulos externos y responder en consecuencia. Es decir, nos permiten captar la información del medio físico que nos rodea.

2.3. Visión artificial

La visión artificial es una disciplina científica que incluye métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina. Tiene como objetivo principal entender la historia detrás de una imagen [3].

2.4. Síntesis de voz

La síntesis de voz es la producción artificial del habla. El sistema computarizado que es usado con este propósito es llamado computadora de habla o sintetizador de voz y puede ser implementado en productos software o hardware [4].

2.5. Diseño del sistema

En la Figura 2 se muestra un diagrama a bloques con los módulos que componen el sistema embebido propuesto. Mientras que en la Figura 3 se tiene un diagrama a bloques que muestra la arquitectura general del prototipo desarrollado.

- El usuario porta el módulo 1.1, encargado de detectar los obstáculos presentes en el camino del usuario.
- El usuario porta un ordenador de bajo coste (Raspberry Pi 3), encargado de llevar a cabo el procesamiento de los algoritmos del sistema y el módulo 1.3, encargado de capturar las imágenes del camino del usuario.
- El usuario porta el módulo 1.2, encargado de dar una señal al usuario de la presencia de un obstáculo en su camino.
- El usuario porta un par de auriculares, por medio de los cuales recibirá el comando de voz generado por el sistema.

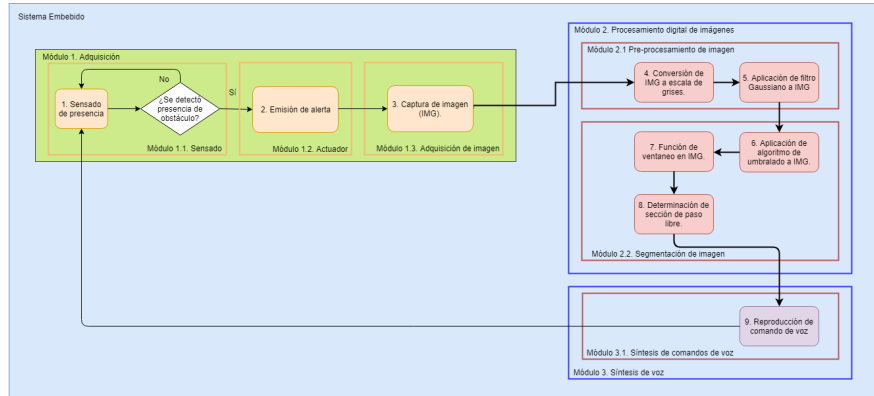


Fig. 2. Arquitectura del sistema embebido.

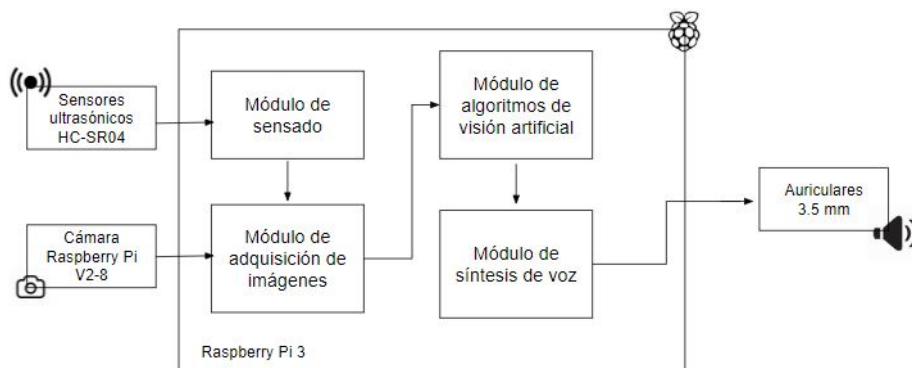


Fig. 3. Descripción general del prototipo.

Módulo 1: Adquisición

Este módulo está encargado de adquirir la información del medio necesaria para el funcionamiento del sistema. Está compuesto por el módulo 1.1. Sensado, el módulo 1.2. Actuador y el módulo 1.3. Adquisición de imagen.

Módulo 1.1: Sensado

En este módulo se realiza la detección de los objetos por medio de la red de sensores, en caso de que se detecte un objeto se procede con la captura de la fotografía para su posterior análisis, en caso contrario se continua sensando.

Módulo 1.2: Actuador

En caso de que uno o más sensores hayan detectado un objeto, se ejecutará este módulo cuyo propósito es activar el actuador que se ha implementado, comunicando al usuario la existencia de un obstáculo en su camino. Se contempla que

el usuario se detenga por 0.5 segundos, permitiendo que la cámara se estabilice y sea capturada la fotografía del camino del usuario.

Módulo 1.3: Adquisición de imagen

Este módulo realiza la adquisición y análisis de imágenes a través de un módulo de cámara para Raspberry Pi modelo V2-8

Módulo 2: Procesamiento digital de imágenes

Conversión RGB a Gray Una imagen RGB consta de 3 capas (rojo, verde, azul), siendo así una matriz tridimensional, por lo que al llevar a cabo esta conversión obtenemos como resultado una representación de la imagen en dos dimensiones, los valores oscilando entre el 0 y el 255 (enteros sin signo de 8 bits). Por lo tanto, los algoritmos aplicados se estarán aplicando a matrices bidimensionales. A través de programación paralela, es posible realizar un procesamiento simple píxel por píxel de una imagen de megapíxeles en milisegundos [5].

Filtro de suavizado Gaussiano El suavizado gaussiano reduce las variaciones de intensidad entre píxeles vecinos, eliminando ruido y detalles finos. La función realiza una convolución de un kernel gaussiano normalizado de 9x9 sobre la matriz de píxeles de la imagen

El kernel de convolución se calcula a partir de la siguiente función, considerando una varianza de 4.0:

$$f(x) = e^{-x^2/s^2}. \quad (1)$$

Módulo 2.2 Segmentación de imagen

Función de umbralado Esta función hace uso del método de umbralado de una imagen, el cual consiste en un proceso de reducción de la información de la misma, sustituyendo todos los píxeles cuya luminancia sea mayor que el coeficiente indicado con el valor de gris 255 (blanco) y sustituyendo todos los demás píxeles por el valor de gris 0 (negro). Para llevar a cabo este proceso se hace uso de un coeficiente de umbral de 50. Una vez que se ha llevado a cabo el proceso de umbralado en las 3 porciones de dimensiones iguales de la imagen, representadas por las etiquetas: “secciónIzquierda”, “secciónCentro” y “secciónDerecha”; dentro de cada una se lleva a cabo un conteo de la densidad de píxeles para determinar la o las secciones que se encuentran obstruidas o libres. Con base en este resultado se genera una instrucción de voz sintetizada por medio de la herramienta eSpeak [6].

Módulo 3: Síntesis de voz

Módulo 3.1 Síntesis de comandos de voz Una vez que ha finalizado la conversión binaria y el conteo de píxeles con valor 0 en cada una de las secciones propuestas, se verifican los valores de los contadores para generar un comando con la herramienta eSpeak.

Se considera que un cuadrante se encuentra obstruido si la densidad de píxeles negros dentro de éste es igual o mayor a 100, pues este valor representa la

cantidad mínima aproximada que un objeto como el descrito en las restricciones del escenario puede contener. Los comandos que reproduce el sistema se muestran en el cuadro 1 con los parámetros considerados para su generación: (Se referirá a los cuadrantes derecho, central e izquierdo con las abreviaciones: **CD**, **CC**, **CI**, respectivamente.)

Tabla 1. Tabla de instrucciones.

CI (px)	CC (px)	CD (px)	COMANDO	DESCRIPCIÓN
<100	<100	<100	"Paso libre, continúe con precaución."	Según los valores obtenidos, el camino se encuentra libre.
>=100	<100	<100	"Lado izquierdo obstruido."	CI está obstruido; CC y CD se encuentran libres.
<100	>=100	<100	"Centro obstruido."	CC está obstruido; CD y CI están libres.
<100	<100	>=100	"Lado derecho obstruido."	CD está obstruido; CC y CI están libres.
<100	>=100	>=100	"Lado izquierdo libre."	CI está libre; CC y CD están obstruidos.
>=100	>=100	<100	"Lado derecho libre."	CD está libre; CC y CI están obstruidos.
>=100	>=100	>=100	"No hay paso libre."	Todos los cuadrantes están obstruidos.

Pruebas Las pruebas correspondientes al módulo 1, nos ayudan a definir la distancia promedio a la cual los sensores detectan los objetos que se encuentran en el escenario.

Con base en la medida del bastón y la altura promedio de una persona a la altura de la cintura, se propone una distancia de 1 m para detectar los objetos, esto se puede observar en la Figura 4 recordando que cada una de las piernas llevara un módulo de adquisición como el mostrado en la Figura 5.

Se realizó una prueba tomando 50 muestras de sensado, adquiriendo la señal en la Raspberry y registrando los tiempos de obtención de respuesta.

Posteriormente se realizó una gráfica, Figura 6, de los datos obtenidos, donde puede observarse que el pico obtenido es de 0.00090 segundos.

Pruebas unitarias del módulo 1.3 - Adquisición de imagen Se encontró que a la altura de 1.20, con una inclinación de 25°, se obtuvieron las fotografías de mayor utilidad.

Una vez que se definió la altura y la inclinación a la que la cámara debe estar posicionada, se realizaron múltiples capturas para asegurar que todos los objetos que son considerados como obstáculos aparezcan en el área adecuada dentro de la fotografía.

Estas pruebas se muestran en las Figuras 7 y 8



Fig. 4. Medición de los objetos.

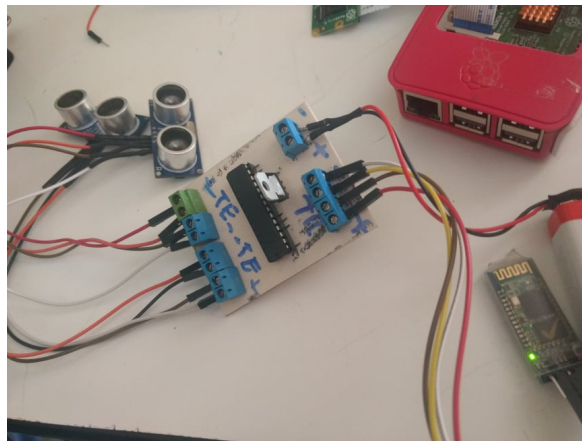


Fig. 5. Placa final del sistema embebido.

2.6. Pruebas unitarias del módulo 2 - Procesamiento de imagen

Una vez que se obtuvieron múltiples imágenes con los ajustes anteriormente mencionados, se seleccionaron aquellas imágenes que cumplen con las condiciones mencionadas en el escenario propuesto: Los objetos considerados como obstáculos serán aquellos con una luminosidad menor a 35 % (color semi-oscuro a oscuro).

Posteriormente, éstas fueron utilizadas como la entrada del algoritmo de procesamiento de imágenes desarrollado e implementado en el proyecto. Ejemplo de entrada es la Figura 9, la salida en la Figura 10 y la instrucción generada en la Figura 11.

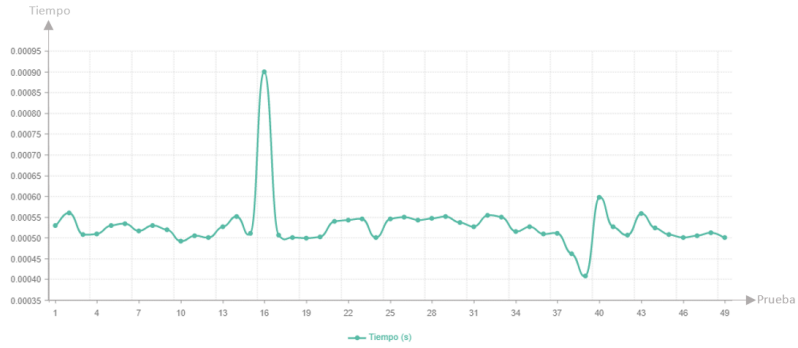


Fig. 6. Gráfica de rendimiento, tiempo de respuesta.



Fig. 7. Pruebas de distancia.

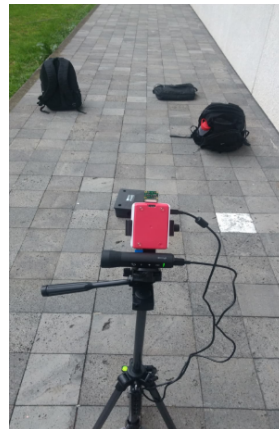


Fig. 8. Pruebas de altura.



Fig. 9. Imagen original.



Fig. 10. Resultado del algoritmo.

2.7. Pruebas de integración

En las figuras 12 y 13 se muestra el prototipo usado durante las pruebas y el modo en el que fue verificado su funcionamiento.

Prototipo de sistema embebido para la asistencia de personas con discapacidad visual en la acera...

```
katia@ubuntu:~/Downloads/TT2V3$ ./principal
Píxeles en centro: 7003
Píxeles a la izquierda: 878
Píxeles a la derecha: 2109
No hay paso libre
```

Fig. 11. Instrucción generada por el algoritmo.

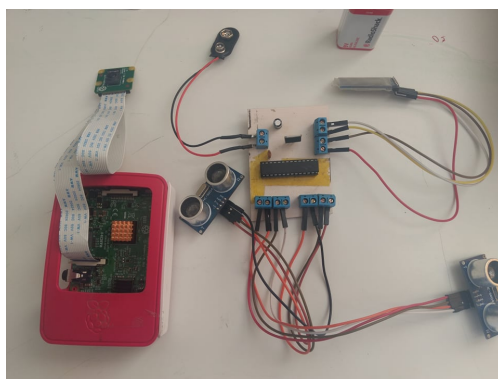


Fig. 12. Prototipo del sistema embebido.

Para las figuras 14, 15 y 16 mostradas anteriormente los comandos recibidos fueron los siguientes:

- Figura 14: "Lado derecho libre".
- Figura 15: "Lado izquierdo obstruido".
- Figura 16: "Lado izquierdo obstruido". .

Al probar el prototipo funcionando en su totalidad, se realizó una prueba, obteniendo 70 muestras que nos permitieron conocer el comportamiento del sistema.

La gráfica de la figura 17 nos permite visualizar el tiempo de ejecución del algoritmo entre el evento en que un objeto es detectado y el comando de voz es reproducido.

Es posible observar que el pico fue de 0.39 segundos.

La gráfica de la figura 18 es posible observar que el pico de utilización de CPU (%) en la prueba fue de 15%.

A continuación se muestra la gráfica de rendimiento donde puede observarse el uso de memoria de la Raspberry, Figura 19, donde se muestra que la memoria permaneció estable, ocupando solamente el 2% de la memoria total.



Fig. 13. Pruebas físicas del prototipo.



Fig. 14. Prueba con un bote de basura.



Fig. 15. Prueba con una mochila.



Fig. 16. Prueba con diversos objetos.

3. Resultados

Se logró desarrollar una red de sensores que detecta el 95.9% de los objetos que se encuentran en el paso del usuario a una distancia de 180 cm, desarrollar e

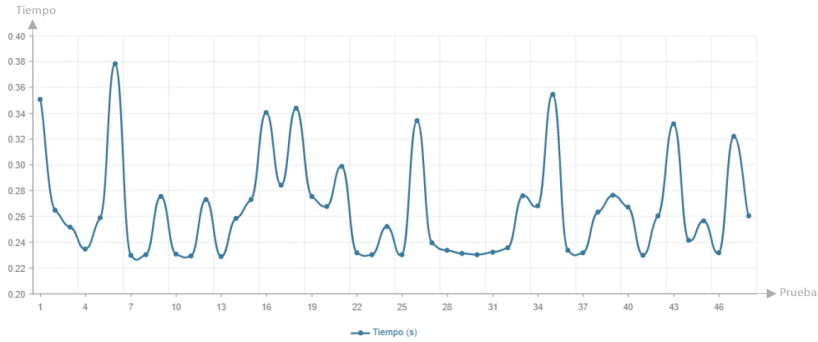


Fig. 17. Gráfica de rendimiento, tiempo de ejecución.

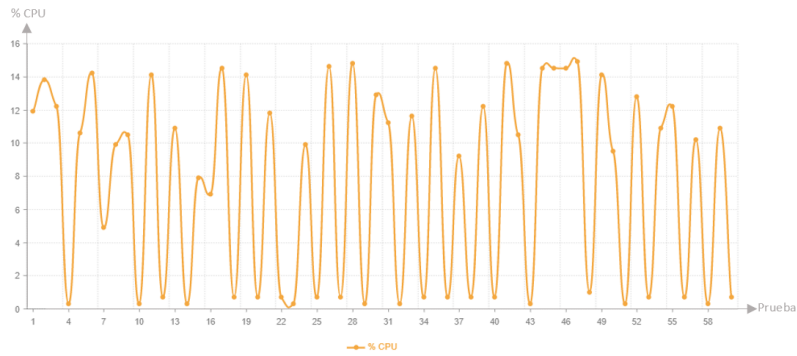


Fig. 18. Gráfica de rendimiento, CPU.

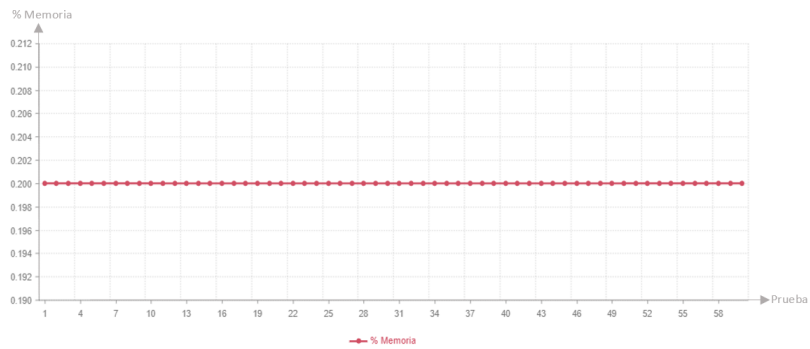


Fig. 19. Gráfica de rendimiento, memoria utilizada.

implementar un algoritmo, paralelizado e implementado en lenguaje C, usando técnicas de visión artificial capaz de determinar las zonas libres y obstruidas

en el camino del usuario a partir de una fotografía capturada por el sistema, generar y comunicar al usuario las instrucciones de asistencia correspondientes a través de la librería *espeak*, en un tiempo máximo de 0.38 segundos a partir de que se ha detectado del objeto, implementar comunicación inalámbrica entre el módulo de sensado y la Raspberry disminuyendo el tiempo total de ejecución del algoritmo en un 16 %.

Fue posible reducir el tiempo de ejecución del algoritmo, permitiendo al sistema comunicar instrucciones al instante.

Con las configuraciones aplicadas, fue posible expandir el escenario controlado, permitiéndonos llevar a cabo pruebas en distintos escenarios y con distintos objetos que cumplieran las especificaciones mencionadas en la definición del escenario.

4. Conclusiones

El diseño del prototipo se realizó con base en los requerimientos y objetivos que se plantearon en un inicio, se logró realizar el prototipo en un tamaño reducido en comparación con los sistemas actualmente existentes. El costo del prototipo es bajo y éste no depende de una conexión a internet para su funcionamiento.

Se logró desarrollar una red de sensores que detecta el 95.9 % de los objetos que se encuentran en el paso del usuario a una distancia de 180 cm.

Se logró desarrollar e implementar un algoritmo, paralelizado e implementado en lenguaje C, haciendo uso de técnicas de visión artificial, capaz de determinar las zonas libres y obstruidas en el camino del usuario a partir de una fotografía capturada por el sistema, emitiendo los comandos de voz pertinentes en tiempo casi real, obteniendo un pico de 0.38 segundos a partir de que un objeto ha sido detectado.

Se logró implementar comunicación inalámbrica entre el módulo de sensado y la Raspberry disminuyendo el tiempo total de ejecución del algoritmo en un 16 %.

Se logró diseñar físicamente el sistema de manera que no interfiera en el paso del usuario y que éste no sea dependiente de una red Wi-Fi.

Gracias a la modularización del prototipo desarrollado, éste puede ser fácilmente escalable, tanto en términos de hardware como de software, pues es posible trasladar el software desarrollado a una tarjeta de desarrollo con dimensiones menores, incorporar distintos sensores y de esta manera, obtener un mayor número de señales que nos permitan tener más información del medio. Por otro lado, es posible desarrollar e implementar nuevas etapas en el algoritmo de procesamiento de imágenes que nos permitan extraer más información de las imágenes para poder generar comandos más detallados.

Agradecimientos. Los autores agradecen a la Escuela Superior de Cómputo y al Centro de Investigación en Computación del Instituto Politécnico Nacional por el apoyo recibido, las herramientas y los conocimientos brindados para el desarrollo del presente trabajo.

Referencias

1. Rodríguez, E.: Diagnóstico sobre la situación de las personas con discapacidad en México. pp. 12–16 (2016)
2. Perez, A., Berreteaga, O., Ruiz de Olano, A.: Una metodología para el desarrollo de hardware y software embebidos en sistemas críticos de seguridad. <http://www.iiisci.org/journal/CV%24/risci/pdfs/C863GM.pdf> (2011)
3. Sossa, H., Morales, R: Procesamiento y análisis digital de imágenes. pp. 20–25 (2012)
4. Black, A., Taylor, P:, Síntesis de voz: Conversión texto a fonemas. <https://francocarlos.com/category/material-academico/> (2017)
5. Cook, J.: Three algorithms for converting color to grayscale. <https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/> (2009)
6. Solé, L.: Binarización de las imágenes. <http://www.dimages.es/Tutorial%20A.I/segmentacion/binaria.htm/> (2018)

Ramath: Mobile Application for Math Learning using Augmented Reality

Hiram David Cortés Díaz¹, Octavio Elías Piñal Ramírez¹,
Amadeo José Argüelles Cruz¹, Claudia Marina Vicario Solórzano²

¹ Instituto Politécnico Nacional, Centro de Investigación en Computación,
Mexico City, Mexico

² Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería y
Ciencias Sociales y Administrativas, Mexico City, Mexico
hcortesd1200@alumno.ipn.mx, opinalr1200@alumno.ipn.mx, jamadeo@ipn.mx,
cvicario@ipn.mx

Abstract. This paper presents a mobile application (app) using Augmented Reality (AR) with the support of game play mechanics in a non-game application, oriented to help 12 to 15 year's old students to learn several math subjects. We associate concepts such as gamification and Problem-Based Learning (PBL) within the app, making the subjects much more interesting to the students and encouraging them to be fearless of math by taking advantage of the human's psychological predisposition to engage in gaming. With the collected information from the research and the help of a predictive algorithm, we try to show how much the technology can help in education.

Keywords: augmented reality, mathematics, gamification, mobile application, learning tool, predictive model, education.

1 Introduction

"When will I use math?" is a question often posed by students from elementary to high school, wondering how topics they see in school will play a role in the everyday life. In recent years there is a tendency in most of the students in the world, they consider subjects like mathematics and science hard, complex and unnecessary as they can't see how they could use them. Because of that, careers oriented on these subjects seem to have a low demand, which could be a problem in the development of science [1].

There are tests designed to measure the level of knowledge of the students in the basic level (elementary – high school) in maths and Spanish. These tests are a good starting point to analyze the performance in these subjects and what can we do to help students improve. According to the Organization for Economic Co-operation and Development (OECD) only 1% of the 15 years old Mexican students have good grades in math and science [2]. According to Plan Nacional para las Evaluaciones de los Aprendizajes (PLANEA) only 6.3% of all Mexican students have an acceptable performance in mathematics [3].

It seems right to use interactive activities that, with the help of technology, can help and allow a better understanding of maths so students can get attracted and see how they can be used outside the classroom. Augmented reality (AR) and problem-Based Learning (PBL) approach could make a good fit to design the lessons and the way to navigate through them, also, gamification could help us to attract the students, making the lessons more engaging and interesting.

The remainder of this paper is structured as it follows. In section 2 we mention and describe some works similar to ours, commenting their main characteristics. In section 3 we define some of the concepts relevant to our work and specify how and where we apply it, as well as the specifications of the tests. Section 4 presents the results of the tests performed. Finally section 5 concludes and gives the highlights of the findings.

2 Related Work

The scope of this paper is the development of an AR app for educational purposes in math subjects. Previous works have already considered the use of AR in this field. For example, Kaur et al. [5] developed an AR web-based application with the proposal of teaching 3D geometry concepts and improve the visualization abilities of middle school students. The study of Hanafi et al. [6] suggest that the AR could improve the motivation in learning subjects because of the lack of confidence and practice. Similar implementation to Ramath is the work of [7], their results suggest that the use of AR generates enthusiasm in the students and also remarks the accessibility and ease of use of the application.

The relationship of AR and education is addressed in the work of Hsin-Kai Wu et al. [8] they suggest that the AR could help learners to engage in an authentic exploration of the real world, and virtual objects such as texts, videos, 3D models and pictures are supplementary elements for learners to conduct investigations of the real-world surroundings. Second, the use of AR technologies can extend to integrating real-world and digital learning resources. The idea is not to replace traditional learning but use AR as a complementary tool.

Meta-Analysis studies were conducted between 1992 and 2011 to compare the effects of games and simulations with traditional classroom instruction based on students' performance. The work of Dimitrov et al. [9] comprises 67 studies and their major findings were that 56% of the experiment population showed no difference between games and conventional instruction but 32% favoured games, while 5% favoured conventional instruction. The analysis made by Wolfe et al. [10] comprises 7 studies; they found out that games are rated as more interesting than conventional instruction. Game-based approach produced significant knowledge-level increases over the conventional case-based teaching methods.

In the work of Hays et al. [11] they analyzed 105 different studies, their major findings were that an instructional game will only be effective if it meets specific instructional objectives and used as it was intended. Instructional games should be embedded in instructional programs that include debriefing and feedback. The use of instructional support help learners to understand how to use the

game increases instructional effectiveness of the gaming experience. At the same time, instructional designers are needed to design games.

Vogel et al. [12] analyzed 32 studies, they found that higher cognitive gains were observed in subjects using interactive games versus traditional teaching methods (although simulations yielded a stronger result). Better attitudes toward learning when compared to traditional teaching methods. Level of picture realism in the computer program does not seem to have an impact.

Ke et al. [13] made an analysis of 600 studies and they showed that the effects of learning with games was positive in 52% of the studies examined. Stizmann et al. [14] found out, after analyzing 65 studies, that confidence with games 20 percent higher. Declarative knowledge was 11% higher for trainees taught with simulation games. Procedural knowledge was 14% higher with simulation games. Retention was 9% higher with simulation games.

These studies show that the use of games can help to improve learning, we can use these ideas in math, because of the abstract nature of this subject using augmented reality as a visual representation of the contents present a good chance to implement it. AR can be involved in adding complementary information to texts, books or any material used in classrooms or to add interactivity between the students and the object of study and enriches the educational environment.

3 Materials and Methods

Ramath is an AR-based mobile application for visualization of theoretical and practical information about three specific subjects: Geometric figures, Pythagoras theorem and operations with fractions. Concepts and problems are represented with dialogue globes and 3D figures. Each topic begins with a problem that needs to be solved. To solve the problem, the student needs to go through every lesson that explain a specific concept to solve the initial problem. After seeing every lesson, the student will try to solve the initial problem, with the option of going back into the lessons to consult any concept.

3.1 Augmented Reality

AR is the technology that enriches the real world with digital information and multimedia, such as 3D models and videos, overlaying in real-time the camera view of your smartphone, tablet, PC or glasses [15].

Ramath uses QR codes as targets that users need to focus with the mobile's camera, so it will pop the information and 3D figures associated to it, as seen in Fig.2. Each target has a specific lesson of a specific subject, for example, there's a target that shows how to calculate the volume of a pyramid. Once the information and 3D figures has been popped, the user can move freely in the space to visualize the information and figures in different angles.

3.2 Problem-Based Learning

Problem-Based Learning (PBL) is a teaching method, with the use of complex real-world problems as the vehicle to promote student learning of concepts and principles as opposed to a direct presentation of facts and concepts. Besides course content, PBL can promote the development of critical thinking skills, problem-solving abilities, and communication skills [16].

In Ramath, we use PBL to define the structure of the lessons and the order of appearance. In every subject, we show the main problem that needs to be solved, as seen in Fig. 1. After that the app shows all the concepts, formulas and some examples that are needed to solve the problem, as seen in Fig. 2. Finally the main problem is showed but now with some answers that we can choose, as seen in Fig. 3. In the case that the answer is incorrect, we can try again or return to some of the concepts, formulas and examples.

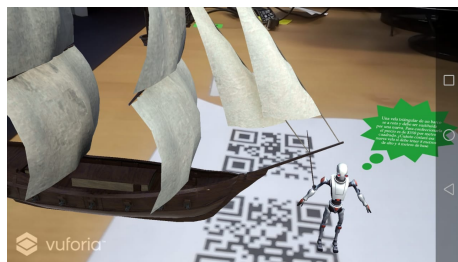


Fig. 1. Main problem to solve.

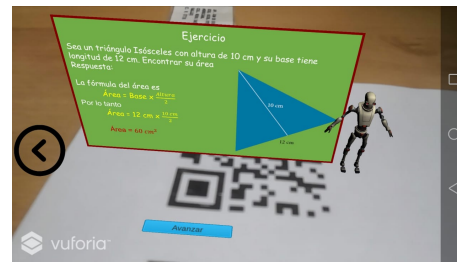


Fig. 2. Concepts.



Fig. 3. Trying to solve the main problem.

3.3 Gamification

Karl Kapp defines gamification as "the use of game play mechanics for non-game applications in order to encourage people to adopt the applications. It also

strives to encourage users to engage in desired behaviors in connection with the applications. Gamification works by making technology more engaging, and by encouraging desired behaviors, taking advantage of human's psychological predisposition to engage in gaming. The technique can encourage people to perform chores that they ordinarily consider boring, such as surveys, shopping, or reading websites" (Kapp Karl, *The Gamification of Learning and Instruction*, 2012) [17] To understand mathematics, the brain needs a positive attitude, so it can learn about it and use that knowledge in other areas of people's lives aside the classroom. Gamification could be an adequate tool for this purpose, using game mechanics in the app, we can make theoretical subjects much more fun and interactive, and we can make more users to be attracted by it.

3.4 Research Design

To measure the impact of Ramath we designed two groups: Control and experimental group. We collected data from 20 participants chosen randomly, 10 of them were part of an experimental group, and the other 10 were part of the control group, all of them belong to Mexico City and were 14 years old at the moment of the test, see Fig. 4. They solved an exam that consisted on 11 questions related to the subjects selected to make a comparison between the grades of both groups. The control group solved the exam without using the app. The experimental group, interacted with the app and used the contents to solve the exam.



Fig. 4. Control and Experimental group.

The variables identified in this case study were

- Dependent Variable: Student's grades.
- Independent Variable: Ramath's use.
- Quantitative Variable: Grades from the test exam.

Statistic Analysis We used linear Regression Model to make a prediction of how much Ramath's users can raise their grades in school. It allows to find the

expected value of a random variable “a” when “b” takes a specific value [18]. This model has the problem that every result obtained from it will be linear, but grades from the students are never linear, so we opted to use Holt’s trend method to get a much more precise model.

Holt’s trend method Holt (1957) extended simple exponential smoothing to allow forecasting of data with a trend. This method involves a forecast equation “(1)” and two smoothing equations, one for the level “(2)” and one for the trend “(3)” [19]. In “(1)” and “(2)” C_x it’s the average, α and β are the smoothness parameters, Y_x the value of the sample. In “(3)” R_x it’s the expected value.

$$C_x = \alpha Y_x + (1 - \alpha)(C_{x-1} + b_{x-1}), \tag{1}$$

$$b_x = \beta(C_x + C_{x-1}) + (1 - \beta)b_{x-1}, \tag{2}$$

$$R_x = C_x + b_x. \tag{3}$$

4 Results

In Fig. 5 and Fig. 6 we show the results of both groups. By grades, we mean the last 3 bimonthly grades they obtained in math. We can observe that there is a difference of 1.8% between the groups in the average of the test results, we can assume that the difference is due to the use of Ramath in a controlled environment.

Like for mathematics	Like for Videogames	1 grade	2 grade	3 grade	Test
Yes	No	7.4	7.6	9	4.5
Yes	Yes	8.3	10	10	4.5
No	No	10	7.9	10	4.5
Yes	Yes	9.1	8	9.5	4.5
Yes	No	8.2	10	9.5	3.6
Yes	Yes	7.1	6	8.9	3.6
Yes	Yes	9.6	8.1	9.8	3.6
Yes	Yes	8.2	6.6	10	3.6
Yes	Yes	7.1	8.4	9.1	2.7
Yes	Yes	9.2	8.6	8.8	2.7
Average		8.42	8.12	9.46	3.78

Fig. 5. Experimental group results.

Like for Mathematics	Like for Videogames	1 grade	2 grade	3 grade	Test
No	No	10	7.6	8	6.3
No	No	8.2	7.8	6	6.3
No	No	9.2	10	9.1	4.5
No	No	7.2	6.5	9	4.5
No	Yes	8.4	7.3	7.8	3.6
No	No	9	10	9.8	3.6
Yes	Yes	7.6	7.4	9.2	3.6
Yes	Yes	10	6.3	7	3.6
Yes	Yes	9.5	9.2	9.4	0
No	Yes	7.2	7	7.5	0
Average		8.63	7.91	8.28	3.6

Fig. 6. Control group results.

Like for Mathematics	Like for Videogames	1 grade	2 grade	3 grade	Test
No	No	10	7.6	8	6.3
No	No	8.2	7.8	6	6.3
No	No	9.2	10	9.1	4.5
No	No	7.2	6.5	9	4.5
No	Yes	8.4	7.3	7.8	3.6
No	No	9	10	9.8	3.6
Yes	Yes	7.6	7.4	9.2	3.6
Yes	Yes	10	6.3	7	3.6
Yes	Yes	9.5	9.2	9.4	0
No	Yes	7.2	7	7.5	0
Average		8.63	7.91	8.28	3.6

Fig. 7. Control group results.

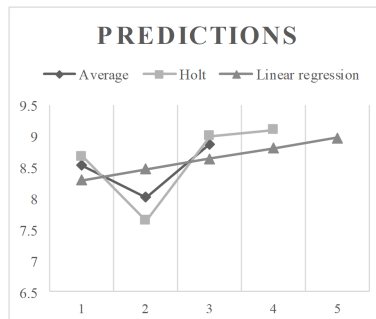


Fig. 8. Comparison of the predictions.

5 Discussion and Conclusions

After the tests we applied the predictive models to the whole data, not distinguishing between the groups. In Fig. 7 we show the average grades of all the participants with the improvement of the 1.8% derived from the experiment. We used the grades as data entry for the linear regression and Holt predictive model. With this, we could predict grades up to 2 bimester. In Fig. 7 and Fig. 8 we can see the results of the linear regression model, the Holts predictive model, the comparison of both predictive models and the average grade of all students.

There is an improvement of about 1.8% between the two groups that made the test. The experimental group was the one with the best results. The only significant difference between the two groups was the use of the app, we assume that the improvement was because of Ramath.

At this point of the research we can not affirm or deny if there is a correlation between the like of video games or mathematics and the results in the test or in the students grades. We will use the predictive models to keep track of students activities, so they will know their performance and to have a better idea of whether prolonged use of the application could improve student grades.

After the experiment we can conclude that in a controlled environment technology can help students to rise their grades and have a better understanding of maths, the experiment moves to a phase 2 in which we will work with a larger number of students and in a prolonged period, expecting to improve and replicate the results obtained.

Acknowledgment. The authors of the present paper would like to thank the following institutions for their economical support to develop this work: Consejo Nacional de Ciencia y Tecnología (CONACYT), SNI, Instituto Politécnico Nacional (COFAA, SIP, CIDETEC, and CIC) and Escuela Secundaria Técnica Número 47 “Juan de Dios Batiz”.

References

1. Randel, J.M., Morris, B.A., Wetzel, C.D., Whitehill, B.V.: The effectiveness of games for educational purposes: a review of recent research. *Simulation and gaming* 23(3), 261–276 (1992)
2. OCDE: Programa para la Evaluación Internacional de Alumnos (PISA) Resultados, 14. Retrieved from <https://www.oecd.org/pisa/PISA-2015-Mexico-ESP.pdf> (2016)
3. Educación Básica 2016 Subsecretaría de Planeación, Evaluación y Coordinación Planea Básica 2016 Informe cobertura. (n.d.). Retrieved from: http://planea.sep.gob.mx/content/general/docs/2016/Aplicacion_PLANEA_Basica.pdf
4. Klopfer, E., Squire, K.: Environmental detectives: the development of an augmented reality platform for environmental simulations. *Educational Technology Research and Development* 56(2), 203–228 (2008)
5. Kaur, N., Pathan, R., Khwaja, U., Murthy, S.: GeoSolvAR: Augmented Reality Based Solution for Visualizing 3D Solids. In: 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), pp. 372–376 (2018)
6. Hanafi, H.F., Said, C.S., Wahab, M.H., Samsuddin, K.: Improving Students' Motivation in Learning ICT Course With the Use of A Mobile Augmented Reality Learning Environment. *IOP Conf. Ser. Mater. Sci. Eng.* 226(1), 012114, Aug. (2017)
7. Lucas, P., Vaca, D., Dominguez, F., Ochoa, X.: Virtual Circuits: An Augmented Reality Circuit Simulator for Engineering Students. In: 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), pp. 380–384 (2018)
8. Wu, H.K., Lee, S.W.Y., Chang, H.Y., Chong Liang, J.C.: Current status, opportunities and challenges of augmented reality in education. *Computers & Education*, Volume 62, pp. 41–49, ISSN 0360-1315, <https://doi.org/10.1016/j.compedu.2012.10.024>. (<http://www.sciencedirect.com/science/article/pii/S0360131512002527>) (2013)
9. Dimitrov, D.M., Rumrill Jr, P.D.: Pretest-posttest designs and measurement of change. *Work* 20(2), 159–165 (2003)
10. Wolfe, J.: The effectiveness games in strategic management course work. *Simulation & gaming* 28(4), 360–376 (1997)
11. Hays, R.T.: The effectiveness of instructional games: A literature review and discussion. Naval Air Warfare Center Training Systems Division (No 2005-004) (2005)
12. Vogel, J.J., Vogel, D.S., Cannon-Bowers, J., Bowers, C.A., Muse, K., Wright, M.: Computer gaming and interactive simulations for learning: A meta-analysis. *Journal of Educational Computing Research* 34(3), 229-243 (2006)
13. Ke, F.: A qualitative meta-analysis of computer games as learning tools. In: Ferdig, R.E. (Ed.), *Effective electronic gaming in education*, Vol. 1, pp. 1–32 (2009)
14. Stizmann, T.: A meta-analytic examination of the instructional effectiveness of computer-based simulation games. *Personnel Psychology* 64(2), 489–528 (2010) [Also: Stizmann, T., Ely, K. A meta-analytic examination of the effectiveness of computer-based simulation games. ADL Research Lab (2011)]
15. Conversus: ¿Qué es la realidad aumentada? Retrieved from: <http://www.cedicyt.ipn.mx/RevConversus/Paginas/RealidadAumentada.aspx> (2016)

16. Dirección General de Docencia de Pregrado: Aprendizaje Basado en Problemas, 8. Retrieved from <http://www.sistema.itesm.mx/va/dide/inf-doc/estrategias/> (s.f.)
17. Kapp, K.M.: The Gamification of Learning and Instruction. In: Davis, M., Zelenko, M., Taff, R. (Eds.), San Francisco, CA. Pfeiffer (2012)
18. Abuín, J.R.: Regresión lineal múltiple. Estadística, Editor 32. In: Nicole, R.: The Last Word on Decision Theory. J. Computer Vision (2007)
19. Taylor, J.W. Exponential smoothing with a damped multiplicative trend. International journal of Forecasting 19(4), 715–725 (2003)

Selección óptima de cartera de inversión de stocks de S&P 500: Uso de algoritmos genéticos

Marco Adrián Guido Torres, Volodymyr Ponomaryov

Instituto Politécnico Nacional, SEPI de ESIME Culhuacan,
Ciudad de México, México

marco@guidotorres.mx, vponomar@ipn.mx

Resumen. El primer modelo de optimización de carteras de inversión fue propuesto por el doctor Markowitz en 1952 y es conocido como modelo de varianza promedio, es usado para encontrar la asignación optimizada de recursos entre diferentes activos financieros, obteniendo retornos de inversión mayores y menores riesgos de inversión. Sin embargo, el modelo de Markowitz no consideraba algunas limitaciones prácticas de los mecanismos de los mercados disponibles, entre ellos: I) costo de transacción, cantidad mínima de reasignación de recursos. Sin estas consideraciones, la funcionalidad del modelo se ve limitada, y diferentes enfoques se han utilizado para intentar resolver el problema, en este artículo se ataca usando algoritmos genéticos.

Palabras clave: portfolio, mean-variance model, genetic algorithms.

Optimal Selection of S&P 500 Stock Investment Portfolio: Use of Genetic Algorithms

Abstract. The first model of optimization of investment portfolios was proposed by Dr. Markowitz in 1952 and is known as the average variance model, it is used to find the optimized allocation of resources between different financial assets, obtaining greater investment returns and lower investment risks. However, the Markowitz model did not consider some practical limitations of the available market mechanisms, including: I) transaction cost, minimum amount of reallocation of resources. Without these considerations, the functionality of the model is limited, and different approaches have been used to try to solve the problem, in this article it is attacked using genetic algorithms.

Keywords: portfolio, mean-variance model, genetic algorithms.

1. Introducción

La forma óptima de selección de portafolios propuesta inicialmente por Markowitz [1] en 1952, utilizaba un enfoque de análisis estadístico clásico, para encontrar la

cartera de inversión óptima, pero no consideraba limitaciones prácticas relacionadas al costo de recolocación de recursos y de la cantidad mínima que se puede reasignar. Si se consideran estos factores, la complejidad del problema aumenta y sus restricciones cambian considerablemente, lo que se suma a un problema que ya es de complejidad NP, como se demuestra en [4].

Múltiples algoritmos heurísticos han demostrado un buen desempeño al atacar este tipo de problemas y [5], [6] y [7] lo hacen usando algoritmos genéticos, reinforcement learning y redes neuronales respectivamente, obteniendo buenos resultados que a su vez.

En este artículo se usa un algoritmo genético para extender el modelo de Markowitz y se aplica a los stocks que componen el índice S&P 500, NASDAQ y FTSE y se compara con el modelo estándar.

2. Descripción del problema

Un portafolio (o cartera) es un grupo de activos financieros como acciones, bonos y fondos en lo general, los portafolios son administrados por profesionales financieros, y su propósito es reconstruir continuamente un portafolio de inversión de acuerdo con el riesgo asociado de los activos incluidos y los objetivos de corto y largo plazo.

Para determinar una asignación de recursos adecuada entre los activos incluidos en un portafolio, se debe considerar el riesgo individual de cada activo y el riesgo asociado equivalente del grupo seleccionado, el portafolio elegido debe ser construido considerando una compensación entre el riesgo y el retorno de inversión asociado. Un portafolio será mejor que otro si tiene una mejor expectativa de ganancia y un menor riesgo asociado.

La teoría moderna de portafolios asume que los inversionistas son muy renuentes al riesgo; si dos portafolios ofrecen el mismo retorno esperado de inversión, típicamente se seleccionará el que tenga un menor riesgo, sin embargo, para que exista un motivo para arriesgarse por un portafolio ligeramente más riesgoso, las expectativas de retorno tienen que ser considerablemente mayores.

Además, si la correlación entre un activo con otro es alta, entonces el riesgo de que ocurran pérdidas generalizadas es mayor, por lo que es importante seleccionar activos con un bajo nivel de correlación para mantener el nivel de volatilidad controlada.

3. Modelo de Markowitz

El modelo de Markowitz se puede resumir a partir de las ecuaciones:

1. El retorno esperado de cada portafolio se define como:

$$E(R_p) = \sum_i w_i E(R_i) \quad (1)$$

$$\sum_i^n w_i = 1 \quad (2)$$

Donde:

- R_p representa el retorno del portafolio completo.
- R_i es el retorno específico de cada activo.
- w_i es la importancia relativa de cada activo. relativo al conjunto de recursos disponibles.
- n representa la cantidad de activos disponibles.

2. La volatilidad del portafolio de interés se define como:

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_{ij} \quad (3)$$

Donde:

- σ_{ij} es la covarianza del retorno esperado de los activos,
 - σ_p^2 es la volatilidad y es proporcional al riesgo esperado del portafolio.
3. Teniendo en cuenta que (1) representa el retorno esperado y (3) el riesgo asociado de un portafolio específico, se puede plantear (4) como el modelo que se debe maximizar para obtener el máximo rendimiento con el mínimo riesgo posible:

$$\begin{aligned} & \text{maximizar} \\ & (1 - \lambda) \sum_i^n w_i R_i - \lambda \sum_i^n \sum_j^n w_i w_j \sigma_{ij} \end{aligned} \quad (4)$$

Donde: λ representa la aversión al riesgo deseada y está limitada por $0 < \lambda < 1$.

4. Para maximizar el retorno de inversión considerando los costos de transacciones, es necesario tomar en cuenta el costo de reasignación de recursos y la cantidad de recursos recolocados:

$$\begin{aligned} & \text{maximizar} \quad ((1 - \lambda) \sum_i^n w_i R_i - \\ & \lambda \sum_i^n \sum_j^n w_i w_j \sigma_{ij} - \sum_i^n c \Delta w_i) \end{aligned} \quad (5)$$

Donde:

- c representa el costo de transacción y lambda,
- Δw_i representa el peso reasignado del índice específico.

A partir de este modelo modificado, se puede observar que la función de evaluación del desempeño del portafolio debe estar dado por la ecuación 5.

4. Algoritmos genéticos

Los algoritmos genéticos están inspirados en la selección natural, y son usados para optimizar y buscar soluciones de problemas a través de operadores bioinspirados, mutación, reproducción, selección, etcétera.

A continuación se presentan las características específicas del algoritmo usado.

4.1. Generación de población inicial

Al representar el peso de una señal, la cual se trata como un gen y al considerar todos los genes como un vector de pesos, un vector compuesto de pesos constituye un cromosoma que representa un portafolio.

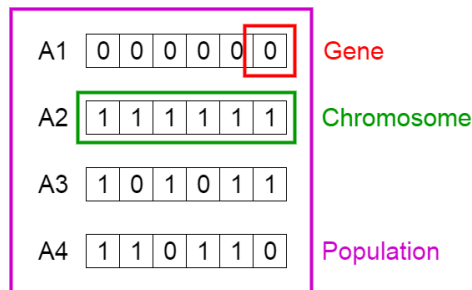


Fig. 1. Representación de la población inicial.

Como población inicial para este algoritmo se presentan 100 cromosomas de longitud variable, dependiendo de la cantidad de stocks que compongan el índice analizado, para S&P 500 se usan 500 genes, como se representa en la figura 1.

4.2. Evaluación

Para determinar una calificación del desempeño de cada individuo de la población creada, se usa una función adecuada de evaluación propuesta por la ecuación 5, además, como función de evaluación extra se asigna el nivel de retorno esperado dentro de un umbral específico móvil que asegura la generación completa de la frontera de eficiencia, es por tanto una función de evaluación multiobjetivo.

4.3. Selección

De la población completa se selecciona a los individuos que hayan obtenido mejor puntuación en la función de evaluación y algunos individuos al azar que a su vez pasaran a un registro histórico llamado salón de la fama, para este caso se usa un sistema de selección basado en “rueda de la fortuna.”

4.4. Recombinación

Dos puntos de cruce son seleccionados aleatoriamente (Fig. 2) de cada pareja de padres, los cuales intercambian las cadenas de genes entre ellos y producen la siguiente

generación, la probabilidad de cada pareja de padres de tener crías es comúnmente proporcional al desempeño de cada padre y sustituyen a los individuos peor evaluados de la siguiente generación.

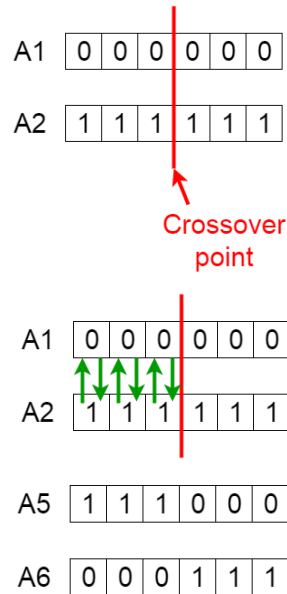


Fig. 2. Recombinación de genes sencilla.

4.5. Mutación

Cada integrante de la nueva generación tiene una probabilidad de que sus genes sean alterados con una probabilidad muy baja, para este caso de .3 con una función gaussiana con promedio en el mismo nivel del valor del gen específico y $\sigma=1$ (Fig. 3).

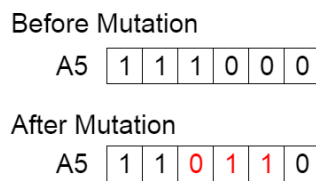


Fig. 3. Mutación sencilla para genes binarios.

4.6. Terminación

Es necesario crear un criterio de finalización, el cual, hasta no ser cumplido, implicará la repetición de todos los pasos anteriores, lo que creará una generación por cada iteración hasta obtener resultados satisfactorios, el criterio del presente algoritmo es la repetición hasta no mejorar el algoritmo en 10 iteraciones consecutivas. En la Fig. 4 se presenta el esquema general del algoritmo genético:

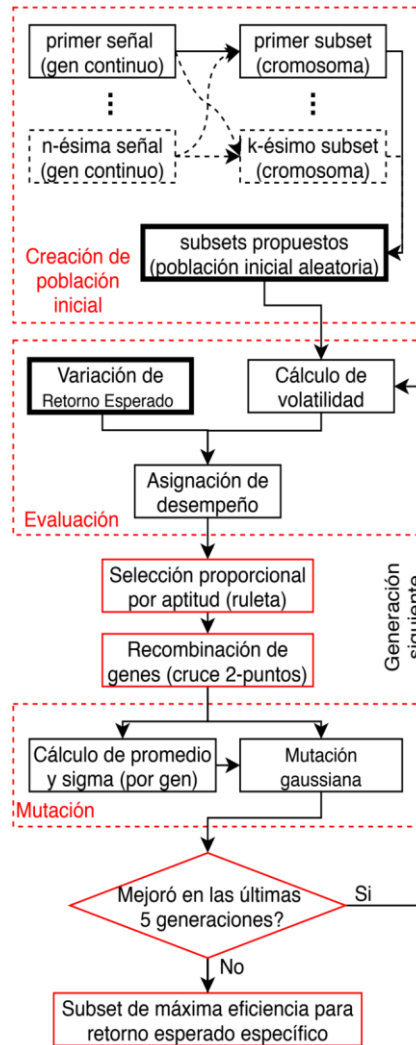


Fig. 4. Diagrama general de algoritmo genético.

5. Resultados

La frontera de eficiencia se dibuja a sí misma al intentar obtener las combinaciones de señales donde la volatilidad sea mínima en distintos niveles de retorno esperado, este borde contiene a los mejores conjuntos de señales e inutiliza todo lo que se aleje de él. No sería apropiado elegir combinaciones con el mismo nivel de retorno esperado pero con mayor riesgo o el mismo nivel de riesgo pero con menor retorno esperado.

Dependiendo del nivel de aversión al riesgo, cualquier punto (combinación obtenida) ubicado en la frontera de eficiencia que se encuentre entre los triángulos azul (mínimo riesgo) y rojo (máximo retorno) puede representar una opción válida de

selección de señales dentro del universo de configuraciones posibles.

En la Fig. 5 se observan los resultados de ejecutar el algoritmo para 10 señales de stocks incluidos en S&P 500.

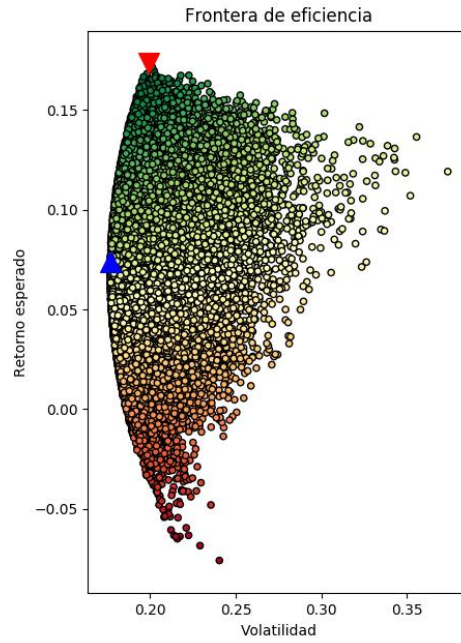


Fig. 5. Frontera de eficiencia obtenida para S&P 500.

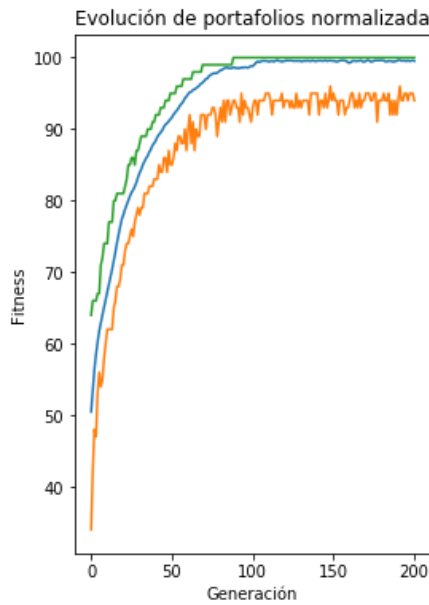


Fig. 6. Evolución normalizada de portafolio para riesgo determinado.

6. Conclusiones

Se propuso un método para explorar las combinaciones posibles de un conjunto de señales financieras con resultados satisfactorios, se observa una aproximación correcta a la frontera de eficiencia por lo que se cumple el objetivo.

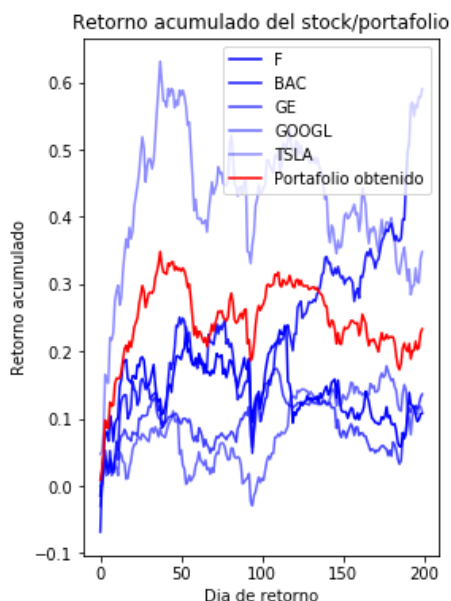


Fig. 7. Retorno de inversión de distintos stocks y portafolio propuesto.

Referencias

1. Markowitz, H.: Portfolio selection. *The Journal of Finance*, vol. 7, pp. 77–91 (1952)
2. Sharpe, W.F.: The sharpe ratio. *The journal of portfolio management*, vol. 21, pp. 49–58 (1994)
3. Li, B., Hoi, C.H.: Online portfolio selection: A survey. *ACM Computing Surveys*, vol. 46 (2014)
4. Potter, M.A., De Jong, K.A.: Cooperative Coevolution: An Architecture for Evolving Co-adapted Subcomponents. *Evolutionary Computation* 8(1), 1–29 (2000)
5. Pereira, R.: Genetic Algorithm Optimization for Finance and Investments. MPRA Paper University Library of Munich, pp. 1–26 (2000)
6. Fama, E.F.: Market efficiency, long-term returns, and behavioral finance. *Journal of Financial Economics*, vol. 49, pp. 283–306 (1999)
7. Chen, Y., Yang, B., Abraham, A.: Flexible neural trees ensemble for stock index modeling. *Neurocomputing*, vol. 70, pp. 4–6 (2007)
8. Demiguel, V., Garlappi, L., Uppal, R.: How Inefficient is the 1/N Asset-Allocation Strategy? (2005)
9. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* 6(2), 182–197 (2002)

Sistema de monitoreo de consumo eléctrico con interfaz para teléfono inteligente

Alejandro Andrés Serapio Carmona¹, Ismael Díaz Rangel²,
Rodolfo Zola García Lozano¹, Cuauhtémoc Hidalgo Cortés¹

¹ Universidad Autónoma del Estado de México, Centro Universitario Ecatepec, México

² Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Aragón, México
andreserapio@hotmail.com, zolagarcia@yahoo.com,
chidalgoc@uaemex.mx, ismael1099@hotmail.com

Resumen. El consumo eléctrico, es la cantidad de energía eléctrica que se ha utilizado a lo largo de un periodo definido, por la cual se debe pagar. El uso de la energía eléctrica está determinado por el estilo de vida de cada usuario, así como de la actividad para la que se destine su uso [1]. La empresa encargada de proveer la energía eléctrica cuenta con un sistema de cobro de tarifas que se incrementa a mayor consumo [2]. Esta información, se le da a conocer al usuario hasta que termina el periodo y se aplica el cobro. En este trabajo se describe el desarrollo de un sistema, cuya función principal es ser una herramienta que ayude a los usuarios a conocer fácilmente el estado del consumo eléctrico de manera anticipada a la fecha de corte; de esa manera, ellos podrán tomar las medidas pertinentes para utilizar racionalmente la energía eléctrica. Adicionalmente, el sistema brindará un estimado de la cantidad a pagar por el consumo realizado desde el inicio del corte hasta el momento actual. El módulo de monitoreo se comunica con un teléfono inteligente por medio de bluetooth, el control es realizado por un microcontrolador; para la medición de la corriente eléctrica, se utiliza el sensor ACS712 de efecto hall, el cual fue caracterizado para la medición de corriente alterna; además, se guarda un histórico de las mediciones para su análisis.

Palabras clave: consumo eléctrico, monitoreo, sensor de corriente.

Power Consumption Monitoring System with Smartphone Interface

Abstract. Electricity consumption is the amount of electrical energy that has been used over a defined period, for which it must be paid. The use of electrical energy is determined by the lifestyle of each user, as well as the activity for which its use is intended [1]. The company responsible for providing electricity has a system for charging fees that increases to higher consumption [2]. This information is made known to the user until the end of the period and the payment is applied. This work describes the development of a system, whose main function is to be a tool that helps users to easily know the state of electricity consumption in advance of the cutoff date; that way, they can take the appropriate measures to rationally use electric power. Additionally, the system will provide an estimate of the amount to be paid for the consumption made from the

beginning of the cut until the current time. The monitoring module communicates with a smartphone via bluetooth, the control is carried out by a microcontroller; for the measurement of electric current, the hall effect ACS712 sensor is used, which was characterized for the measurement of alternating current; In addition, a history of the measurements is kept for analysis.

Keywords: electricity consumption, monitoring, current sensor.

1. Introducción

La energía eléctrica es un recurso que ha brindado apoyo para el desarrollo de las sociedades. Cuando el uso de la energía eléctrica comenzó a brindar mayores comodidades a las personas, su uso se fue incrementando; tanto, que en la actualidad la energía eléctrica se encuentra en casi todas las actividades diarias de las personas: en el transporte, en la preparación de la comida, en la iluminación, comunicaciones, entretenimiento, trabajo, etc. Su uso es tan intenso, que se ha tenido la necesidad de buscar e invertir en tecnologías que produzcan energía eléctrica a partir del uso de recursos renovables, con la finalidad de poder abastecer la creciente demanda de energía eléctrica, pues al crecer, lo hacen también las actividades relacionadas a su producción; aun así, es necesario cuidar de los recursos energéticos que se utilizan para la producción de energía, por lo que los usuarios deberían utilizar eficientemente la energía eléctrica proporcionada [3]. Para que el usuario tenga información y con ello tomar medidas de gestión del uso de energía, este trabajo propone el desarrollo de una herramienta para informar a los usuarios en cualquier momento, el estado del consumo energético. Adicionalmente, el sistema proporciona al usuario información acerca de la tarifa en que se encuentra, el precio a pagar por dicha tarifa, y el tiempo restante para la fecha de corte. La interfaz está desarrollada en una aplicación móvil, que se enlaza fácilmente con el módulo de monitoreo (o adquisición) mediante bluetooth.

2. Estado del arte

Existen trabajos, que han propuesto una solución para conocer el estado del consumo eléctrico a través del uso de medidores digitales. En [4] se describe un prototipo de medidor electrónico que considera las pérdidas de energía eléctrica ocasionadas por robo de energía a través del uso de sensores de corriente, colocados en segmentos específicos del tendido eléctrico; adicionalmente, el sistema puede establecer una comunicación a través de internet con la empresa que suministra el servicio, donde envía información acerca del estado del consumo; además, se puede activar o desactivar el suministro de energía de manera remota. El sistema utiliza un microcontrolador como sistema de procesamiento y de control, una pantalla LCD para mostrar la información al usuario y un Ethernet Shield para establecer una comunicación a través de internet. El autor refiere que las pruebas realizadas mostraron valores que se pueden considerar aceptables. Una desventaja de este prototipo, es el tendido que se debe realizar para enviar la información por internet a través de una red alámbrica; además, en caso de que el usuario no cuente con una conexión a internet, él tendrá que encargarse de cubrir este servicio. Otro aspecto a resaltar, es que el usuario para poder conocer el estado del

consumo, debe de ir hasta donde se encuentra el sistema para visualizar la información en la pantalla.

Un sistema que posee como objetivo brindar información acerca del estado del consumo antes de que llegue la fecha de corte, se encuentra descrito en [5], el trabajo describe el desarrollo de un sistema de monitoreo basado en un microcontrolador como dispositivo de procesamiento, un medidor de energía utilizando una bobina, un ESP-8622 como medio de comunicación inalámbrica WiFi, y una interfaz web para visualizar la información obtenida. Una desventaja que presenta el sistema es la necesidad de contar con una conexión a internet todo el tiempo; por lo que, si el usuario no cuenta con servicio de internet, no podrá visualizar la información del consumo en la página web.

Por otro lado, el sistema propuesto cuenta con una interfaz de teléfono inteligente que se comunica con el módulo de monitoreo a través de bluetooth, por lo que no se necesita contar con internet; de esta manera, el usuario podrá obtener mediante una app información del consumo, estimado del monto a pagar y días restantes para la fecha de corte. Adicionalmente, la interfaz de la app permite configurar fecha de corte y montos de tarifas. Las mediciones son almacenadas en una memoria MicroSD a razón de 5 segundos para que el usuario pueda revisar el histórico de su consumo.

3. Tarifas

En México la Comisión Federal de Electricidad (CFE) es quien suministra la energía eléctrica, y el esquema de cobro está definido la Comisión Reguladora de Energía (CRE), quien determina la metodología del cálculo de las tarifas para el suministro básico (Artículo 139 de la Ley de la Industria Eléctrica) [1]. Para la Ciudad de México y zona metropolitana el esquema de cobro se muestra en la tabla 1.

Tabla 1. Tarifas aplicadas a CDMX y zona metropolitana.

Consumo	Precio x KWh (MXN)	Rango[KWh]
Consumo básico	\$0.81	0 < 75
Consumo intermedio	\$0.98	75 < 140
Consumo excedente	\$2.87	140 < 250
Alto Consumo Doméstico	\$107 (fijo) + \$4.79	> 250

Para dar un contexto, si se tienen dos cargas de 100 W (como un par de luminarias) durante las 24 horas durante un mes, el consumo total es de **144 KWh** (200 W x 24 h x 30 días), por lo cual se deberá pagar (según la tabla 1) lo siguiente: (75 KWh x \$0.81) + (65 KWh x \$0.98) + (4 KWh x \$2.87) = **\$135.00**

Si se rebasa el rango de 250 KWh, se aplica el cobro de tarifa doméstica de alto consumo (DAC); tomando el ejemplo anterior, pero con cuatro cargas de 100W, el consumo mensual sería de 288 KWh por lo que se debe pagar: \$107 + (288 KWh x \$4.79) = **\$1486.50**, lo que equivale a 10 veces más el monto con respecto al caso anterior. De ahí la importancia de tener la información referente al consumo de manera fácil y puntual, y con ello elegir las medidas pertinentes para gestionar el consumo.

4. Metodología

El sistema propuesto está conformado por varias etapas que se pueden apreciar en el diagrama de bloques de la figura 1. En la figura 2 se presenta el diagrama esquemático del sistema.

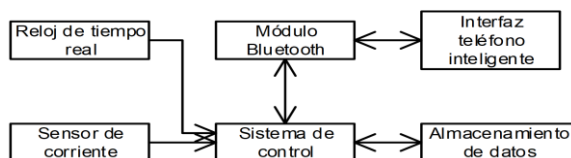


Fig. 1. Diagrama de bloques.

Sistema de control: Es el microcontrolador que se encarga de realizar todas las funciones lógicas del sistema, se utiliza un microcontrolador Atmega328 [6].

Sensor de corriente: Se utilizó el sensor de corriente de efecto Hall ASC712 versión para 30 A, este bloque obtiene la información de la corriente utilizada, la cual se almacena en una variable [7].

Almacenamiento de datos: La información de fecha, hora y consumo, es almacenada en una memoria MicroSD, utilizando un MicroSD shield. Extrayendo la memoria, el usuario podrá revisar el histórico de su consumo, abriendo el archivo generado mediante un programa de hoja de cálculo.

Módulo Bluetooth: Realiza la comunicación con un teléfono inteligente para enviar y recibir datos, utilizando un módulo HC05 [8].

Interfaz para teléfono inteligente: Es una aplicación desarrollada para dispositivos Android mediante appInventor2 [9], la cual es la interfaz visual que permite comunicar al usuario y al sistema.

Reloj de Tiempo Real: De este componente se obtiene la información de la fecha, sirve para mantener un registro por fecha y hora de los valores obtenidos, utilizando un módulo RTC DS1307 [10].

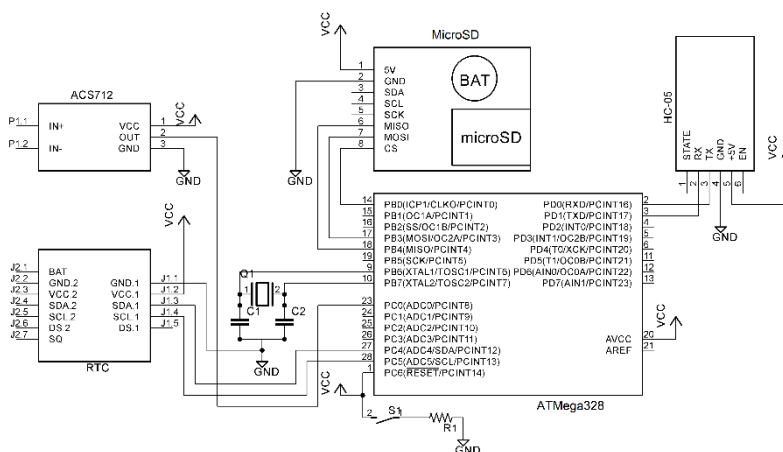


Fig. 2. Diagrama esquemático.

4.1. Sensor de corriente ACS712

Es un sensor de corriente de efecto Hall, es de tipo invasivo, su funcionamiento consta de cuantificar la corriente a través de sus terminales (IN+, IN-), y utiliza el efecto Hall para producir en la salida un voltaje proporcional a la corriente que se encuentra circulando.

El valor en voltaje de la salida se observa en la figura 3, y muestra el comportamiento lineal del sensor.

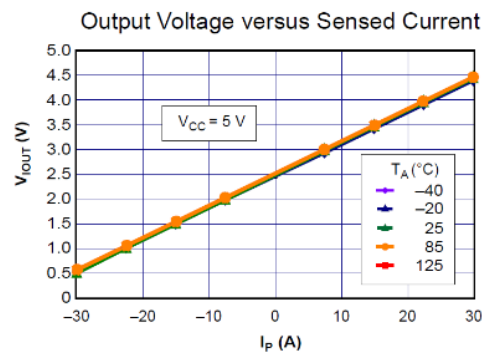


Fig. 3. Gráfica de voltaje vs corriente del ACS712.

4.2. Medición de la corriente eléctrica

Para la obtención del consumo, se realizaron experimentos para caracterizar el comportamiento de la salida del sensor en AC (Corriente Alterna). Para la caracterización, se analizó la salida del sensor con la ayuda de un osciloscopio. Aplicando una carga de 2300 W se obtuvo la señal mostrada en la figura 4, donde se puede apreciar que entrega un voltaje en directa, pero presenta variaciones que van desde un voltaje mínimo 880 mV a un voltaje máximo 4.08 V.

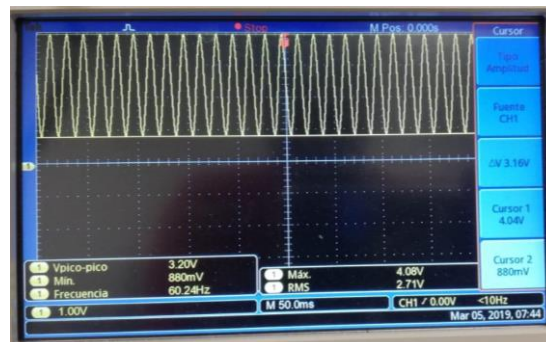


Fig. 4. Señal de salida del sensor con carga 2300W alimentada con AC.

Para calcular el valor de la corriente, se realizó un experimento en el que se conectaron diferentes cargas resistivas, y conociendo la linealidad del sensor, se empleó la ecuación de la recta para aplicar un análisis quedando definida como:

$$V = mI + B, \tag{1}$$

donde:

V = Voltaje de salida del sensor.

I = Corriente que circula por el sensor.

B = Voltaje cuando la corriente es cero.

m = pendiente o razón de cambio, estando definido por:

$$m = \frac{V_2 - V_1}{I_2 - I_1}. \tag{2}$$

De los valores mostrados por el fabricante y verificados experimentalmente, se extrajeron valores que se sustituyen en 2 y se obtiene:

$$m = \frac{4.5-2.5}{30-0} = \frac{2}{30} = \frac{1}{15}. \tag{3}$$

Ahora sustituyendo el valor de m en (1) y utilizando la ordenada al origen, encontramos la expresión:

$$V = \frac{1}{15}I + 2.5. \tag{4}$$

Para obtener la potencia consumida es necesario trabajar con valores eficaces (RMS). Para obtener el voltaje RMS que entrega el sensor, se necesita obtener el voltaje de pico (Vp) máximo de un ciclo de la senoidal, por lo que se implementó un algoritmo que toma lecturas consecutivas durante 17 milisegundos con el convertidor analógico digital (dado que es la duración de un ciclo completo de una señal de 60 Hz), y cada lectura realizada es comparada con la anterior, con el fin de obtener el valor mayor, el cual representa al Vp. Para el cálculo de la corriente RMS por definición se utiliza:

$$I_{RMS} = \frac{I_p}{\sqrt{2}} \quad \text{despejando } I_p = I_{RMS} * \sqrt{2}. \tag{5}$$

Sustituyendo (5) en (4) e indicado en lugar V al Vp, tenemos:

$$V_p = \frac{1}{15}(I_{RMS} * \sqrt{2}) + 2.5 \quad \text{despejando } I_{RMS} = \frac{15(V_p-2.5)}{\sqrt{2}}. \tag{6}$$

Para calcular la energía consumida de manera instantánea (que en realidad será la potencia consumida en un ciclo de la senoidal) se utiliza:

$$P_{ins} = V_{RMS} \times I_{RMS}, \tag{7}$$

donde:

P_{ins} = Potencia instantánea.

V_{RMS} = Voltaje eficaz de la fuente de alimentación.

I_{RMS} = Corriente eficaz obtenida en (6).

Para conocer el acumulativo de energía consumida se calcula con:

$$P_{consumida} = \frac{P_{ins}}{\Delta t}, \tag{8}$$

donde Δt es igual a 3600, se debe a que se definió que cada lectura fuera tomada en intervalos de un segundo. Por la tanto, corresponde a 1/3600 parte de lo consumido en 1 hora.

El algoritmo programado para la carga de valores iniciales (tarifas y fecha de corte) almacenados en la eeprom del microcontrolador, estimación de la potencia instantánea, la transmisión de datos hacia la aplicación se observa en el diagrama de la figura 5.

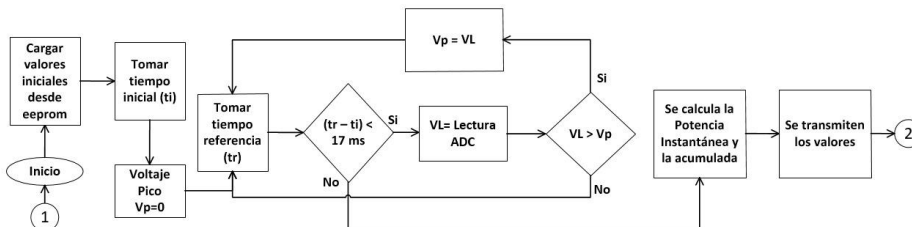


Fig. 5. Diagrama de flujo para obtener la potencia instantánea y su transmisión.

Para poder realizar las lecturas cada segundo, se desarrolló un algoritmo (figura 6) que asegura que haya transcurrido un segundo entre cada lectura, así se toman 3600 lecturas y sumatorias en el intervalo de una hora. Además, en esta sección, se monitorea la recepción de datos provenientes de la aplicación para la reconfiguración de parámetros y su almacenamiento en la eeprom.

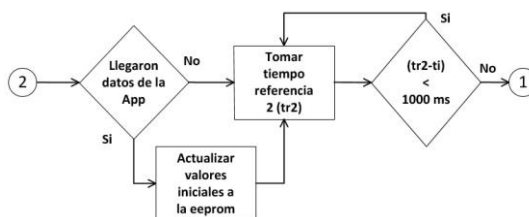


Fig. 6. Diagrama de flujo para tomar lectura cada segundo y reconfiguración de parámetros.

4.3. Aplicación móvil

La aplicación tiene como objetivo servir de interfaz visual y de configuración entre el usuario y el sistema, el funcionamiento de la aplicación se muestra en el diagrama de flujo de la figura 7.

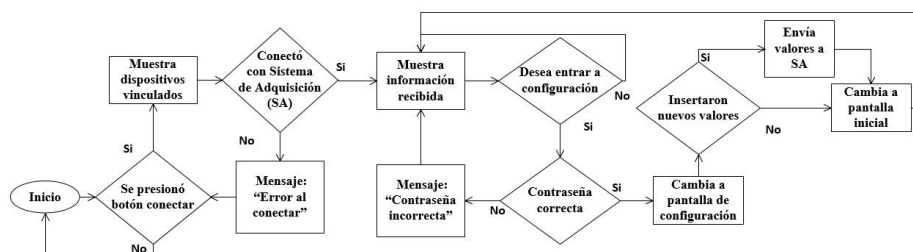


Fig. 7. Diagrama de flujo de la interfaz de la aplicación móvil.

Al ejecutar la aplicación (ver figura 8), la interfaz muestra al usuario la información del consumo, cuánto se va a pagar y cuántos días hacen falta para la fecha de corte.

Para visualizar la información, es necesario pulsar el botón conectar, con el que se enlaza la aplicación con el módulo de monitoreo a través de una comunicación por bluetooth. Cuando se presiona el botón de configuración se solicitará el ingreso de un código de seguridad, con lo que se accederá a la pantalla de la figura 9, los campos corresponden a las tarifas vigentes y la fecha de corte; el botón enviar actualiza la información del sistema, lo cual es importante para establecer la información que se presentará en la pantalla inicial referente al consumo.

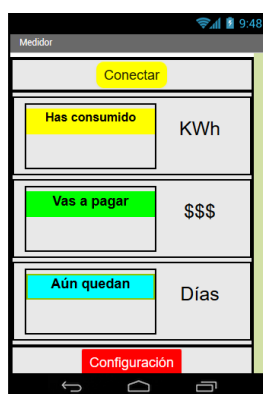


Fig. 8. Pantalla inicial.

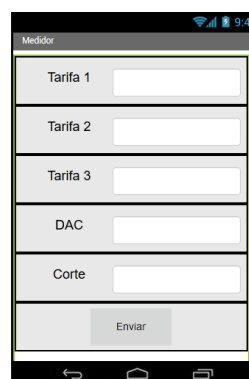


Fig. 9. Pantalla de configuraciones.

5. Pruebas y resultados

A continuación, se muestran los resultados de medir corrientes eficaces sobre diferentes cargas (tabla 2), todas ellas utilizando un voltaje de alimentación nominal de 127 VRMS, utilizando el sistema propuesto (SP) y un multímetro calibrado:

Tabla 2. Resultados de mediciones de corrientes.

Carga	Amperímetro [A]	SP [A]	Diferencia [%]
Lámpara 40 watts	0.336	0.38	14
Lámpara 100 watts	0.58	0.6	4.5
Plancha 1100W	8.65	8.48	-1.9
Estación de soldar	5.82	5.46	-6.5
Plancha 2300W	17.21	16.99	-1.25
Parrilla eléctrica	8.65	8.36	-3.25
Promedio*:			-1.68

* Diferencia promedio sin considerar cargas pequeñas

Se pudo comprobar que para cargas superiores a 0.3 Amper los resultados son bastante aproximados, no tanto así para cargas inferiores; al ser un sistema para su implementación en hogares, no se considera una situación relevante, ya que los consumo suelen ser muy superiores a 0.3 A. Además, se debe considerar que existen fluctuaciones en la red de energía de la compañía proveedora y que eso también es

causa de las diferencias, aunque se trató de realizar las lecturas de manera simultánea, ese factor de error es a considerarse.

Para las pruebas de consumo instantáneo, se conectó el sistema propuesto en el centro de cargas de un hogar en serie con un sistema comercial [SC] de medición de consumo eléctrico con la capacidad de recolección de mediciones. Las mediciones se realizaron durante 7 días, obteniéndose casi tres mil muestras, la figura 10 muestra los resultados comparativos de los sistemas. Los resultados entregaron un valor promedio de mediciones de potencia instantánea para el sistema propuesto de 3.72 W, y para el sistema comercial de 3.7235 W, lo que implica una diferencia aproximada del 0.1 %.

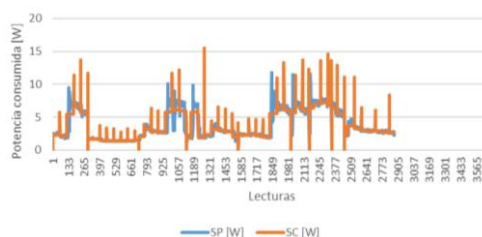


Fig. 10. Comparativa de mediciones.

Para determinar la equivalencia estadística, se aplicó a los datos el método de concordancia propuesto por Bland-Altman [11], el cual muestra una gráfica (figura 11) que indica niveles de correlación entre dos métodos de medición que observan un mismo fenómeno.

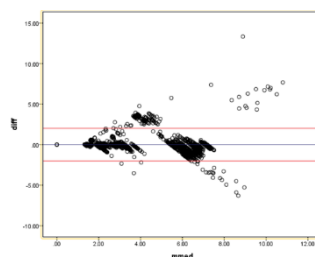


Fig. 11. Gráfica de Bland-Altman de las mediciones entre los dos sistemas.

Se puede observar como la gran mayoría de los datos se encuentran entre las líneas rojas (intervalo de confianza), lo cual permite determinar que los sistemas estadísticamente son equivalentes en sus mediciones.

Tabla 3. Resultados prueba medición consumo.

Fecha	CFE [KWh]	SP [KWh]
17/06/2019	1	1
18/06/2019	0	0
19/06/2019	1	1
20/06/2019	1	1
21/06/2019	0	0
24/06/2019	1	1
25/06/2019	1	1

Para el consumo acumulado, se conectó el sistema a la salida del centro de cargas de un hogar, y se realizaron lecturas de manera manual durante 7 días con respecto al medidor de la CFE (mismo horario entre cada día), se calculó el consumo registrado en términos de KWh, ya que es la unidad de medida del medidor, y se compararon con el sistema propuesto, la tabla 3 muestra los resultados.

Se puede observar que los resultados son idénticos, por lo que se comprueba la eficacia del sistema para medir el consumo eléctrico en KWh.

6. Conclusiones

Se desarrolló el sistema de medición con base a un microcontrolador Atmega328, donde se logró la implementación de algoritmos para la adquisición y procesamiento de datos, así como del control de almacenamiento y transmisión de datos. Cumpliendo de manera precisa con las cotas de tiempos necesarias para la adecuada cuantificación de consumo eléctrico.

Además, se diseñó una aplicación para el sistema operativo Android, la cual funciona como interfaz visual y de configuración, para que el usuario pueda observar la información del consumo proporcionada por el sistema, y también permite configurar algunas variables como tarifas y fecha de corte.

Se llevó a cabo el estudio del funcionamiento en AC del sensor de corriente; esto, a partir del análisis de cargas resistivas conocidas alimentadas con AC, se modeló un algoritmo para la interpretación de las lecturas cuando se trabaja con AC, con esa información, se calculó la potencia instantánea y a partir de esta última, se obtuvo el valor del consumo eléctrico, expresado en KWh.

Se realizó un análisis estadístico de la fiabilidad del sistema con respecto a un sistema comercial, obteniendo valores dentro del intervalo de confianza que indican la equivalencia de mediciones entre ambos sistemas, lo que indica que el sistema posee un grado de aproximación similar a un medidor de tipo comercial.

Referencias

1. Comisión Federal de Electricidad (CFE): https://app.cfe.mx/Aplicaciones/CCFE/Tarifas/Tarifas/Tarifas_casa.asp?Tarifa=DACTAR1&anio=2018. (Último acceso: 22 enero 2019)
2. Escoto Castillo, A., Sánchez Peña, L., Pérez Guardián, G.: Hogares y energía eléctrica en México. *Revista Espinhaço*, vol. V, pp. 30–43 (2017)
3. Riera, P.: *Manual de economía ambiental y de los recursos naturales*. Madrid, Parainfo (2005)
4. Gaibor, E.R. et al: Implementación de un prototipo de medidor de energía eléctrica residencial considerando la reducción de pérdidas no técnicas por hurto. *Publicando*, pp. 66–82 (2018)
5. Díaz Pérez, P., Clemente Arce, V.G., Ortiz Muñoz, J.A.: Sistema embebido basado en el monitoreo de consumo de energía eléctrica de electrodomésticos para el control del gasto. *Tecnología Digital* 8(1), 69–80 (2018)
6. Microchip: ATmega328 Datasheet. http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (Último acceso: 14 noviembre 2018)

7. Allegro MicroSystems, Inc., «Allegro MicroSystems,»
https://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwig2OHr9Y_jAhXDXc0KHXINACEQFjABegQIABAC&url=https%3A%2F%2Fwww.allegromicro.com%2F~%2Fmedia%2Ffiles%2Fdatasheets%2Facs712-datasheet.ashx&usg=AOvVaw1LkWwxiA9ZbTsL_UhbCLAp. (Último acceso: 5 octubre 2018)
8. Makezine: Make Magazine. 1 marzo 2014. https://cdn.makezine.com/uploads/2014/03/hc_hc-05-user-instructions-bluetooth.pdf. (Último acceso: 18 febrero 2019)
9. MIT: MIT APP INVENTOR. <https://appinventor.mit.edu/explore/ai2/tutorials.html>. (Último acceso: 24 enero 2019)
10. Maxim integrated: <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>. (Último acceso: 14 noviembre 2018)
11. Altman, D., Bland, J., Altman, D.G., Bland, J.M.: Measurement in medicine: the analysis of method comparison studies. *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 32, n° 3, pp. 307–317 (1983)

Sistema de navegación y evasión de obstáculos aplicando un sistema de control difuso en una placa Arduino UNO

David Jesús Segura Cristino, Herón Molina Lozano, Elsa Rubio Espino,
Víctor H. Ponce Ponce

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México
absol.1994@gmail.com, hmolina@cic.ipn.mx, erubio@cic.ipn.mx

Resumen. La navegación autónoma de robots móviles se ha convertido en un área de gran interés, debido a su presencia en diversas áreas industriales. En este trabajo se propone la implementación de un sistema de control difuso tipo Mamdani que utiliza la defuzificación del Método del Centro de Máximos (COM), en una plataforma con recursos computacionales limitados (arduino UNO); demostrando así, que es posible implementar controles difusos en sistemas de poco poder computacional sin tener que implementar un sistema de comunicación a una plataforma con mayores recursos (computadora).

Palabras clave: conjuntos difusos, lógica difusa, control difuso, robot diferencial, navegación autónoma.

Navigation and Obstacle Avoidance System Applying a Fuzzy Control System on an Arduino UNO

Abstract. Autonomous navigation of mobile robots has become an area of great interest, due to its presence in various industrial areas. This paper proposes the implementation of a Mamdani type fuzzy control system that uses the defuzzification of the Maximum Center Method (COM), on a platform with limited computational resources (arduino UNO); thus demonstrating that it is possible to implement fuzzy controls in systems of little computing power without having to implement a communication system to a platform with greater resources (computer).

Keywords: fuzzy sets, fuzzy logic, fuzzy control, differential robot, autonomous navigation.

1. Introducción

Actualmente la mayoría de las industrias están empleando robots para la realización de tareas, ya que estos poseen precisión y repetibilidad en sus acciones; razón por la cual es de gran importancia el desarrollo constante de nuevos

algoritmos de control que se adapten a los constantes cambios de tecnología y que se cumplan con los requerimientos eficiencia que se necesitan en cualquier tipo de aplicaciones industriales, de medicina, etc. La capacidad de un robot móvil de posicionarse de manera correcta es muy importante para que éste sea capaz de alcanzar un objetivo o que pueda evadir obstáculos de un ambiente industrial o de difícil acceso [1].

Por otro lado, es ya conocido que el control difuso nos permite implementar un control con razonamiento similar al de un ser humano, en un sistema computacional. Es muy útil cuando el sistema a controlar es complejo y no hay un modelo matemático o la implementación de dicho modelo consume muchos recursos, ya que el control difuso no requiere de ningún modelo para su uso. Este tipo de sistemas se pueden mejorar fácilmente pues el cambio de algún conjunto de entrada no afecta a las normas de control o al resto de los conjuntos de entrada/salida [1].

Por lo dicho anteriormente, en este trabajo se propone un sistema de control para la navegación y evasión de obstáculos de un robot diferencial no holonómico (RDNH) utilizando un sistema de control difuso tipo Mamadani.

Este artículo está organizado de la siguiente manera: el estado del arte se presenta en la sección II, el desarrollo teórico en la sección III, el desarrollo experimental se encuentra en la sección IV, seguido de la discusión de resultados en la sección V y finalmente se presentan las conclusiones.

2. Estado del arte

En el pasado se han realizado otros trabajos para comprobar el desempeño de diversos métodos para el control de robots móviles. Por ejemplo, en [3] se propone un sistema de control basado en la función de Lyapunov y para su implementación se tienen que encontrar tanto las variables a controlar, como las ecuaciones de la velocidad lineal y angular del robot, las ecuaciones de las velocidades angulares de los motores derecho e izquierdo y finalmente se usa odometría para conocer la posición del robot. Si se compara lo anterior con un sistema difuso, puede apreciarse que la dificultad de implementación en un sistema difuso es mucho menor.

El trabajo realizado en [1] trata de un sistema de control de navegación parecido al propuesto en este artículo, en el que se usa la lógica difusa y en donde se observa que la implementación del sistema es bastante sencilla, obteniendo buenos resultados en la simulación numérica del sistema. Un sistema de evasión de obstáculos que hace uso de la lógica difusa es [6], el artículo únicamente trata el problema de la evasión de obstáculos, por lo anterior no se implementa ninguna forma de navegación para llegar a un objetivo. En [4] se propone un sistema en chip que usa lógica difusa para navegación, este sistema está montado en un FPGA, por lo que aumenta la velocidad de procesamiento del sistema de control. El trabajo en [7], usa un sistema neuro-difuso para la evasión de obstáculos y esta implementado en una computadora la cual se conecta inalámbricamente con un robot que también le comunica las lecturas de sus

sensores. Una implementación similar a la propuesta en este trabajo, es la presentada en [2]; en donde los comportamientos de evasión y navegación utilizan un control difuso. La implementación del sistema se realizó en un robot Scout-II, que a su vez se conecta inalámbricamente a un *host* WiFi, en él se encuentra montado un servidor con el sistema de control, lo anterior es posible debido a que la velocidad de comunicación es suficiente para que sistema responda oportunamente ante cualquier situación que se presente en el recorrido.

En los trabajos anteriores, la mayoría de los sistemas requieren de un sistema externo para montar el control difuso que se comunica con el robot, aprovechando así la facilidad de programación y los recursos computacionales de los que podría carecer el robot móvil que se controla. A pesar de lo anterior, hay ejemplos en [6], [8], [5], en los cuales se han implementado controles difusos sencillos en placas Arduino las cuales carecen de los recursos de una computadora; pero que se podrían conectar al sistema a controlar mediante WiFi. Tomando como base lo anteriormente dicho, en este trabajo, se usará también una placa Arduino para la implementación del sistema de control difuso propuesto. Ver sección 4.

3. Desarrollo teórico

3.1. Odometría

Ya que se usará control difuso, no se requiere un modelo del sistema, sin embargo, el vehículo requiere conocer su ubicación en todo momento para saber cómo debe comportarse y cómo afectan las decisiones que tome el sistema de control. La odometría permite conocer la posición del robot móvil. Se usará *dead-reckoning* que es un método con base en la posición y orientación previas al tiempo $t = t_k$. Usando una aproximación de Euler en el modelo cinemático del robot diferencia (ver ref.[2]).

$$\begin{cases} X(k+1) = X(k) + v(k)T_s \cos(\theta(k)), \\ Y(k+1) = Y(k) + v(k)T_s \sin(\theta(k)), \\ \theta(k+1) = \theta(k) + \omega(k)T_s. \end{cases} \quad (1)$$

donde $T_s = T_{k+1} - T_k$ es el tiempo de muestreo. Lo más conveniente es usar la distancia que se desplazó el robot durante el tiempo de muestreo, esto puede ser calculado usando un *encoder* para conocer el desplazamiento angular de cada llanta, otro método consiste en inferir dicho desplazamiento usando la velocidad de desplazamiento a la que gira el motor, para esto se requiere que la velocidad a la que gira el motor se pueda controlar, cómo se hace en un servomotor o un motor paso a paso. Por lo anterior se modificó la ecuación 1, quedando como sigue:

$$\begin{cases} X(k+1) = X(k) + \frac{\pi D}{2} (\Delta T_l + \Delta T_r) \cos(\theta(k)), \\ Y(k+1) = Y(k) + \frac{\pi D}{2} (\Delta T_l + \Delta T_r) \sin(\theta(k)), \\ \theta(k+1) = \theta(k) + \frac{\pi D}{2} (\Delta T_r - \Delta T_l). \end{cases} \quad (2)$$

donde $\Delta T = T_{k+1} - T_k$, es la cantidad de revoluciones que realiza una rueda en el tiempo de muestreo (l se refiere a la rueda izquierda y r , se refiere a la rueda derecha).

A grandes velocidades el *Dead Reckoning*, muestra poca precisión en la estimación de posiciones pero a velocidades bajas demuestra un buen desempeño [2]. Este es el método que se uso en el desarrollo experimental de este trabajo para conocer la posición del robot.

3.2. Comportamientos

Para que el robot sea capaz de navegar hasta un obstáculo y además pueda esquivar obstáculos, se requiere que tenga ambos comportamientos programados y sea capaz de alternar entre ambos, para esto se usaron sensores ultrasónicos que permiten reconocer si hay un obstáculo frente al robot y a qué distancia se encuentra el robot de él. Cuando los sensores envían una señal de una presencia en cierto rango activarán el comportamiento que permite esquivar obstáculos del robot y el robot comenzará a moverse de acuerdo a lo dictado por dicho comportamiento y a su vez le permitirá alejarse o evitar el obstáculo. Mientras no haya nada que le impida el paso, el robot estará usando el comportamiento de navegación, que lo lleva a buscar la ruta más corta a un objetivo definido. Alternando entre ambos comportamientos el robot es capaz de desenvolverse en un entorno completamente desconocido. Una descripción de la implementación de ambos comportamientos se puede ver en la figura 1.

3.3. Navegación

El comportamiento de navegación es el que hace que el robot se mueva hacia el objetivo para que el robot cumpla con su misión. Las entradas de este sistema son dos y los vemos representados en la figura 2, la distancia desde la posición del robot hasta el objetivo, la cual se calcula de la siguiente forma:

$$P = \sqrt{(X_r - X_t)^2 + (Y_r - Y_t)^2}, \quad (3)$$

donde X_r, Y_r es el vector de posición del robot, y X_t, Y_t es el vector que describe la posición del objetivo a alcanzar del robot.

La segunda variable de entrada de este control es el error de ángulo α :

$$\alpha = \beta - \theta, \quad (4)$$

donde β es el ángulo del vector de distancia del robot al objetivo y θ es la orientación actual del robot.

La salida del sistema son las velocidades de giro de los motores w_R y w_L que son la velocidad angular de la llanta derecha y la velocidad angular de la llanta izquierda respectivamente. Los motores usados son motores Dynamixel AX-12A los cuales no reciben velocidades angulares, sino valores que van desde -1023 a 1023, por esto la variable de salida no será dada en rad/s , sino en valores con

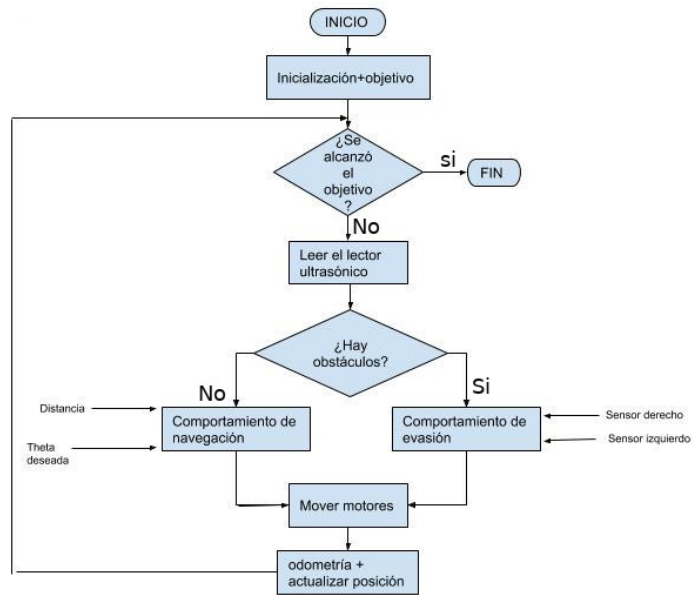


Fig. 1. Diagrama de flujo del comportamiento del robot móvil.

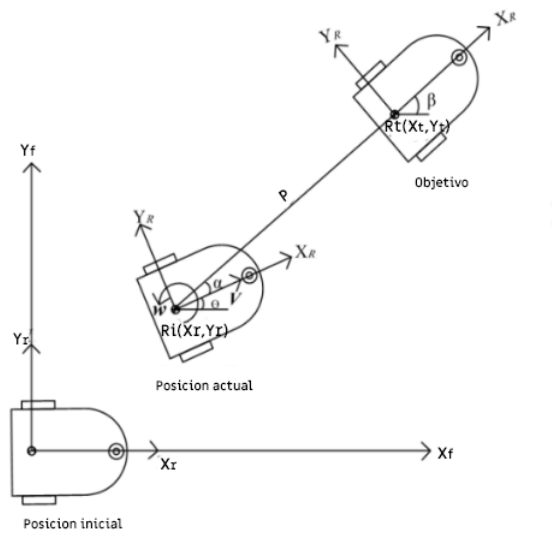


Fig. 2. Distancia al objetivo (P) y error de orientación (α) [3].

el mismo rango al que trabajan los motores. Lo anterior es una ventaja pues no es necesario calcular las salidas en velocidad angular y realizar conversiones de

rad/s al valor numérico que representaría dicha velocidad en los motores, lo cual acarrearía errores que se acumularían en cada iteración del comportamiento.

Se usaron cinco conjuntos difusos para las variables de entrada, esto nos permite tener un mejor control y precisión. Para los conjuntos de la variable P se usaron cuatro conjuntos triangulares para las distancias cercanas al robot móvil y un trapezoidal para el conjunto que representa la distancia más lejana. En cuanto a la variable de control α se usan cinco conjuntos triangulares y su rango de operación va de $-\pi$ a π . En las figuras 3 y 4 se pueden ver las gráficas que muestran éstos conjuntos.

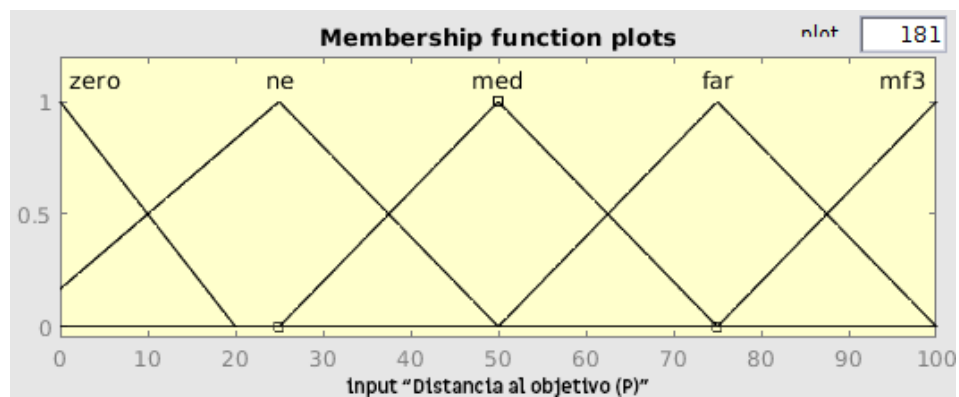


Fig. 3. Distancia al objetivo (P).

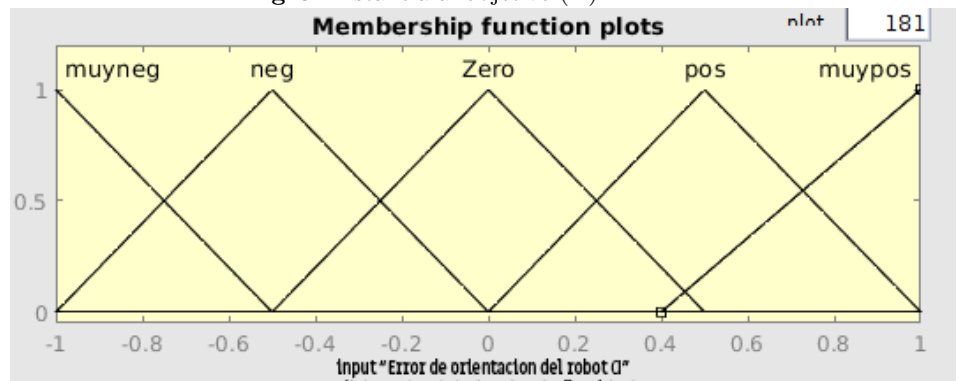


Fig. 4. Error de orientación del robot α .

Como ya se mencionó, las variables de salida no serán dadas en rad/s , sino en valores entre -1023 y 1023. La velocidad a la que se mueven los motores cuando se les da un valor de 1023 es muy alta y podría generar errores, por lo anterior el rango de operación de los motores se reducirá a la mitad. En las figuras 5 y 6 podemos ver cómo son dichos conjuntos.

Ya que tenemos cinco conjuntos de cada variable de entrada, se generan veinticinco reglas difusas *si-entonces* las cuales se muestran en la figura 7.

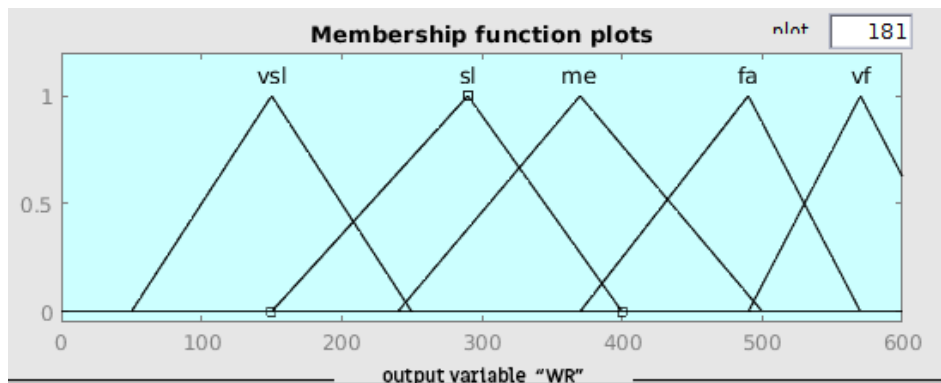


Fig. 5. Velocidad de movimiento del motor Derecho (wr).

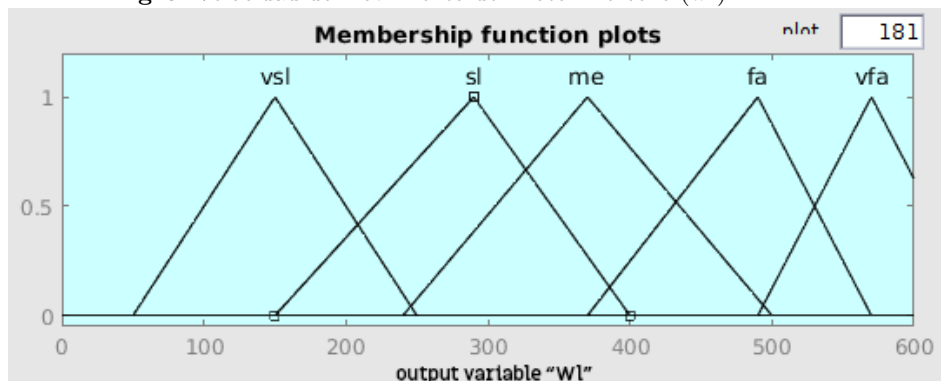


Fig. 6. Velocidad de movimiento del motor Izquierdo (wl).

3.4. Evasión de obstáculos

La evasión de obstáculos por parte del robot, ocurre cuando los sensores ultrasónicos detectan que hay un obstáculo que impide su avance, y que además se encuentra dentro de la ruta generada por el comportamiento de navegación. Sus variables de entrada son las distancias medidas por los sensores ultrasónicos (Dl y Dr). Las distancias están expresadas en *cm* para facilidad del usuario, y sus variables de salida son las velocidades de los motores, los cuales, al igual que en comportamiento de navegación se encuentran entre -600 y 600 para evitar que el robot se mueva velocidades muy altas generando errores en la odometría.

Puesto que en este caso ambas variables tienen el mismo rango de trabajo, pero en posiciones diferentes, se pueden usar los mismos conjuntos difusos para

alphaP	zero	ne	med	far	veryfar
muyneg	Rvsl	Rvsl	Rsl	Rm	Rm
	Lm	Lm	Lf	Lf	Lvf
neg	Rvsl	Rvsl	Rvs	Rs	Rsl
	Lsl	Lsl	Lm	Lf	Lf
Zero	Rvsl	Rsl	Rm	Rf	Rvf
	Lvsl	Lsl	Lm	Lf	Lvf
pos	Rsl	Rsl	Rm	Rf	Rf
	Lvsl	Lvsl	Lvsl	Lsl	Lsl
muypos	Rm	Rm	Rf	Rf	Rvf
	Lvsl	Lvsl	Lsl	Lm	Lsl

Fig. 7. Reglas difusas de navegación.

ambas. Estos son cinco conjuntos trapezoidales que van desde 0 *cm* hasta 200 *cm* y se muestra en las figuras 8 y 9; aunque para mostrarlas correctamente sólo se muestra el rango de 0 a 50 *cm*.

Los conjuntos difusos de las variables de salida se pueden ver en las figuras 10 y 11.

En este caso se generarán nueve reglas difusas que definirán cómo responde este sistema de control figura 12.

3.5. Robot móvil

Para realizar las pruebas con el sistema de control se usó un robot construido con el kit Bioloid (figura 13), disponible en el laboratorio de Robótica y Mecatrónica del CIC, se reemplazó el controlador que venía con el kit Bioloid y en su lugar se usó una tarjeta de desarrollo Arduino UNO.

Los motores empleados para el movimiento del robot son dos motores AX-12A, los cuales se comunican al Arduino mediante comunicación serial, que funcionan de 7v a 12v. Para controlar la velocidad se les debe indicar un número en un rango de -1023 a 1023, mismo que representa los valores de la velocidad máxima de giro del motor.

Para detectar la presencia de obstáculos y la distancia a la que se encuentran estos del robot, se usó el sensor para Arduino HC-SR04.

3.6. Defuzificación

El método de defuzificación más empleado es el método del centroide, pues permite un control más preciso, sin embargo este método implica el cálculo del área de los conjuntos difusos y esto va más allá de las capacidades del Arduino UNO, es por esto que para hacer la defuzificación se usó el método centro de

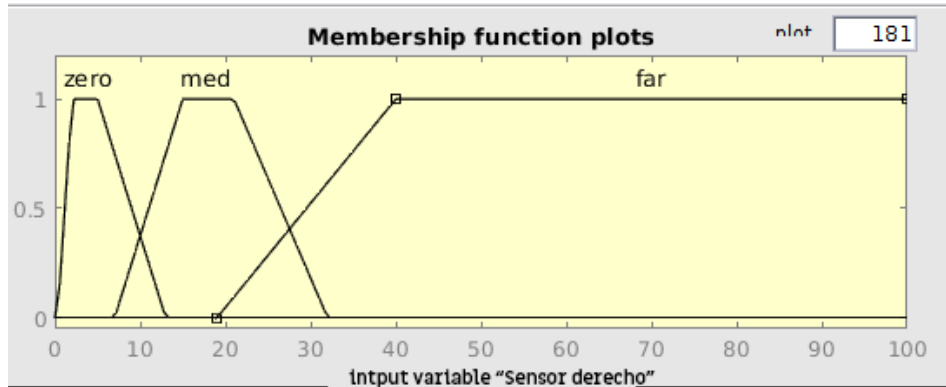


Fig. 8. Distancia al obstáculo sensor derecho.

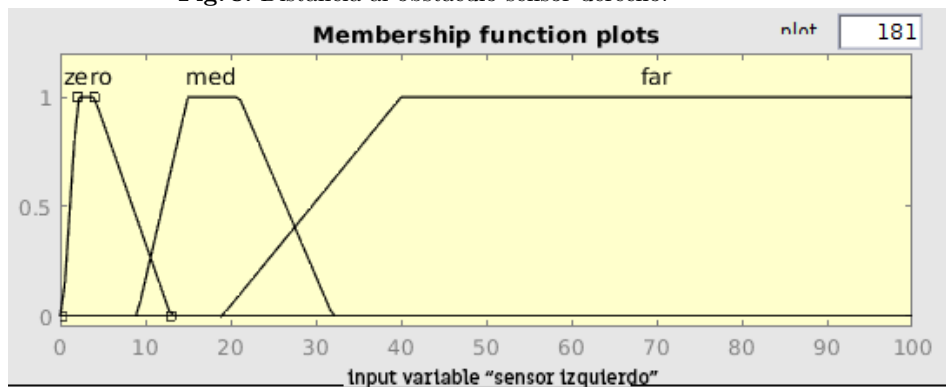


Fig. 9. Distancia al obstáculo sensor izquierdo.

máximos (COM), que se usó en [8], pues es un método que requiere pocos recursos y operaciones algebraicas que fueron programadas en el Arduino UNO.

$$Z^* = \frac{\sum_{i=1}^n \mu(ui) * ui}{\sum_{i=1}^n \mu(ui)}, \quad (5)$$

donde ui es el valor típico de cada set difuso y $\mu(ui)$ es el valor de pertenencia en ese conjunto difuso.

El método COM es eficiente, si se utilizan conjuntos difusos simétricos, además permite obtener salidas reales usando recursos computacionales limitados [8].

4. Desarrollo experimental

Se anexa un link a un video del robot en funcionamiento:

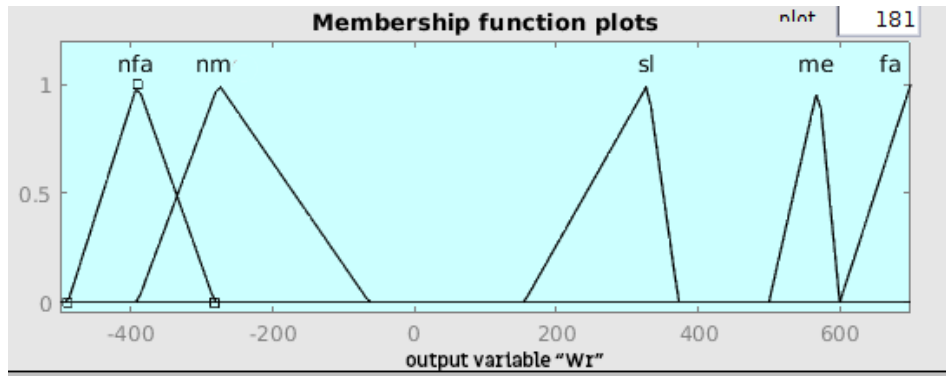


Fig. 10. Velocidad de movimiento del motor Derecho (wr).

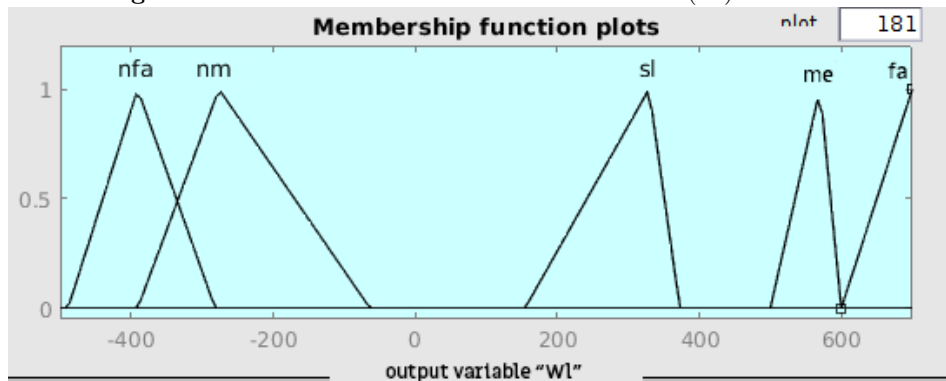


Fig. 11. Velocidad de movimiento del motor Izquierdo (wl).

Dr/Dl	zero	med	far
zero	Rnf	Rsl	Rns
	Lnf	Lnf	Lnf
med	Rnf	Rf	Rme
	Lsl	Lme	Lf
far	Rnf	Rme	Rf
	Lns	Lf	Lf

Fig. 12. Reglas difusas de evasión.

<https://drive.google.com/file/d/1oTFy7lFWEoDNotDydwssSqNKfAXcAcFT/view?usp=sharing>

Se realizaron algunas pruebas con el sistema de control implementado en el robot diferencial con el Arduino UNO (figura 14). En las pruebas el robot

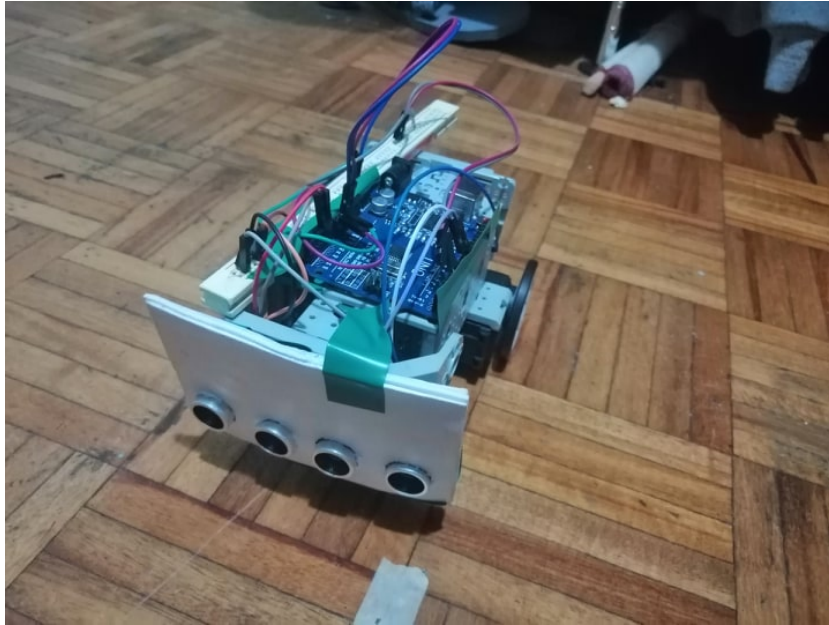


Fig. 13. Robot usado en las pruebas.

fue capaz de llegar a la posición objetivo sin colisiones con los obstáculos y en un tiempo corto (menos de un minuto en las pruebas realizadas). El área en el que se realizaron las pruebas fue en un escritorio del laboratorio de robótica y mecatrónica del CIC, esto debido a las limitaciones de la alimentación del sistema, se usó una fuente de 9v de CD conectada a la toma de corriente.

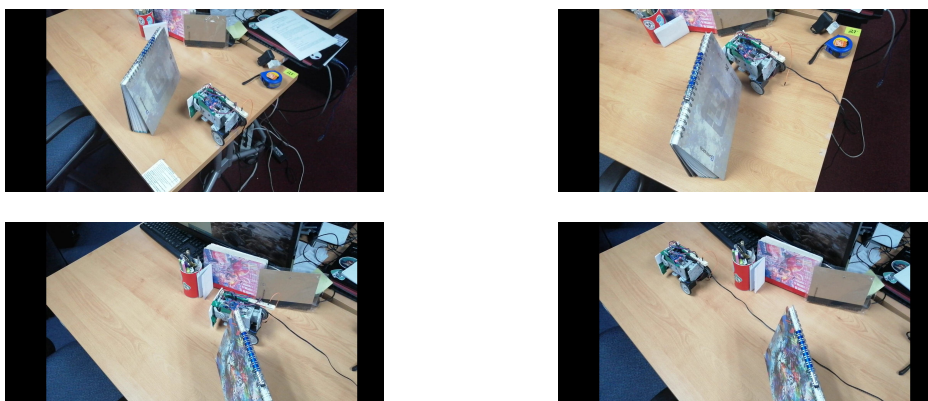


Fig. 14. Imágenes de las pruebas realizadas.

En dos pruebas el robot colisionó con los obstáculos (figura 15(a)) que se encontraban en el área de pruebas. Debido a la ubicación de los sensores ultrasónicos, el robot tiene problemas con los obstáculos que se encuentran fuera del rango de visión de los sensores, pero que están lo suficientemente cerca como para chocar con sus llantas (figura 15(b)), que por el diseño del robot, se encuentran más alejadas del centro de lo que están los sensores.



a) Choque del vehículo con un obstáculo.



b) El Robot móvil tiene problemas con obstáculos muy pegados a sus costados.

Fig. 15. Problemas encontrados durante las pruebas del Robot diferencial.

5. Discusión

Los resultados experimentales demuestran que es posible implementar un sistema de control difuso tipo Mamdani en una tarjeta embebida como el Arduino UNO. Manteniendo los beneficios del uso del control difuso sin usar grandes recursos de *hardware* o de realizar los cálculos del control en una computadora. Todo lo anterior, se reduce sólo a enviarle las variables de velocidad al robot.

Si bien en algunas ocasiones el sistema falló (ver figura 2) y el robot chocó con los obstáculos, ésto se puede llegar a solucionar modificando los conjuntos propuestos o las reglas, lo cual no representa ningún problema, pues la facilidad de mejorar el sistema es una de las ventajas de los sistemas de control difusos.

6. Conclusiones

Si bien los sistemas de control difusos tipo Mamdani pueden llegar a requerir muchos recursos computacionales, es posible y es sencillo el implementarlos en sistemas de bajo poder computacional como un Arduino UNO, si bien se pierde precisión en los cálculos comparando los resultados con un *hardware* con más recursos, ésto se puede compensar modificando la forma de los conjuntos difusos de entrada. Trabajar el control difuso de esta forma permite que no sea necesario implementar un sistema de comunicación inalámbrica o alámbricamente con una computadora, ni cambiar la placa de control por un sistema más poderoso como una Raspberry o una Beaglebone.

Como trabajo futuro se planea refinar este sistema de control modificando los conjuntos difusos, rediseñando el robot, cambiando la ubicación de los sensores y las dimensiones de este.

Referencias

1. Rashid, R., Elamvazuthi, I., Begam, M., Arrofiq, M.: Differential Drive Wheeled Mobile Robot (WMR) Control Using Fuzzy Logic Techniques. In: 2010 Fourth Asia International Conference on Mathematical Analytical Modelling and Computer Simulation, 51–55 (2010)
2. Faisal, M., Hedjar, R., Al Sulaiman, M., Al-Mutib, K.: Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment. International Journal of Advanced Robotic Systems (2013)
3. Sandeep Kumar Malu, B., Majumdar, J.: Kinematics, Localization and Control of Differential Drive Mobile Robot. Global Journal of Researches in Engineering: H Robotics & Nano-Tech, 14(1), 9 (2014)
4. Tzafestas, S. G., Deliparaschos, K. M., Moustiris, G. P.: Fuzzy logic path tracking control for autonomous non-holonomic mobile robots: Design of System on a Chip. Robotics and Autonomous Systems, 58(8), 1017–1027 (2010)
5. Jayetileke, H. R., de Mei, W. R., Ratnayake, H. U. W.: Real-time fuzzy logic speed tracking controller for a DC motor using Arduino Due. In: 7th International Conference on Information and Automation for Sustainability, 1–6 (2014)
6. Bobyr, M. V., Kulabukhov, S. A., Milostnaya, N. A.: Fuzzy control system of robot angular attitude. In: 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 1–6 (2016)
7. Zerfa, H., Nouibat, W.: Fuzzy reactive navigation for autonomous mobile robot with an offline adaptive neuro fuzzy system. 3rd International Conference on Systems and Control, 950–955 (2013)
8. Espino Núñez, A., Ramírez Martínez, F.: Control de temperatura con lógica difusa para un sistema de espectroscopía láser. Tesis de la Facultad de Ingeniería, Universidad Autónoma de México (2017)

Software de estación terrena para cohetes hidropulsados

Alberto Carlos Pérez Cadena, Oswaldo Morales Matamoros,
Diego Alfredo Padilla Pérez, Jesús Jaime Moreno Escobar

Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica,
Centro de Desarrollo Aeroespacial, Ciudad de México, México
jemoreno@esimez.mx

Earth Station Software for Hydropropelled Rockets

Abstract. An embedded system and an earth station software were built for the acquisition and visualization of atmospheric variables, geolocation and physical magnitudes. We used arduino and web programming and websocket js, to create a point-to-point communication between the sensor system and the web application. Finally, an integrated circuit with sensors and a web application with communication was obtained through an XBee.

Keywords: earth station, water rocket, hydropropulsion, embedded system, web development.

1. Introducción

El Instituto Politécnico Nacional a través del Centro de Desarrollo Aeroespacial y equipos académicos miembros del comité organizador Antares Aerospace, IPN Cuauhtémoc y la Asociación Aeroespacial de ESIME Ticomán. Convocaron, a los estudiantes de todas las instituciones educativas pertenecientes a esta casa de estudios, a participar en el Concurso Politécnico de Cohetes hidropulsados 2019.

El concurso busco el desarrollo de un proyecto interdisciplinario, que simule sistemas aeroespaciales complejos. Por un lado, la construcción de un cohete hidropulsado y por otro el desarrollo de un software de adquisición de datos. El software fue un aspecto técnico importante, ya que jugo el papel de una estación terrena, mediante una plataforma de adquisición y visualización de variables en tiempo real basados en módulos embebidos. Se utilizó arduino, obteniéndose un circuito integrado con sensores que median:

- Presión,
- Temperatura,
- Humedad relativa,
- Longitud y latitud mediante GPS,
- Aceleración que sufre el dispositivo en 3 ejes,

- Altitud.

El sistema es una aplicación web con comunicación a través de un XBee. Se uso websockets con javascript, para crear una comunicación punto a punto entre el sistema de sensores y la aplicación web.

Además, se desarrolla el sistema de información bajo un pensamiento sistémico, ya que la sistémica busca crear una sinergia que conjunta soluciones parciales, encontrando sus interrelaciones con la finalidad integrar propuestas como lo son las tecnologías de información, tecnología aeroespacial y sistemas embebidos.

Actualmente, los sistemas para realizar mediciones de variables atmosféricas en estaciones terrenas utilizados en México pueden llegar a ser de alto costo económico, y el acceso al mismo se dificulta debido a que los fabricantes constituyen un mercado reducido. La tendencia actual de utilizar software y hardware de libre distribución, permite tener acceso a dispositivos relativamente sencillos, pero con capacidad de procesamiento adecuado a la implementación de diferentes instrumentos y funcionalidades. Así, hablar de estaciones terrenas y su equipamiento es un tema amplio que requiere un análisis a profundidad, por ello el presente artículo se centra en el software de interfaz de usuario, una interfaz gráfica para realizar procesos de medición de variables y al mismo tiempo un sistema portable y escalable que forma parte del módulo de censado y captura (Payload) para el cohete del concurso.

2. Análisis del entorno de los sistemas actuales de monitoreo

Actualmente existen sistemas de monitoreo de diferentes variables y en lo que respecta al uso de sistemas embebidos e ingeniería de software enfocado a la web hay bibliografía variada reciente y ambos son un tópico estudiado en los últimos años. En la tabla 1 se presenta tres proyectos [8], [9], [10], que involucran tecnologías similares realizados por diversos autores con la finalidad de mostrar aportes en el proceso de investigación y desarrollo de estaciones de monitoreo.

3. Sistemas de hidropulsión bajo un enfoque sistémico

3.1. Ciencia de Sistemas

El pensamiento sistémico no busca solucionar problemas ni proponer soluciones prácticas, más bien es una metodología integradora para el tratamiento de los problemas científicos. Por lo tanto, con un enfoque de sistemas podemos resolver los problemas de un sistema mayor en vez de sistemas componentes, nos enfocamos por optimizar la eficacia del sistema total en lugar de mejorar la eficacia de sistemas cerrados [3].

Tabla 1. Proyectos relacionados con el tema en cuestión.

Identificación	Objetivo	Categoría	Instrumentos	Resultado
Ruiz-Ayala, D. C., Vides-Herrera, C. A., & Pardo-García, A. (2018). Monitoreo de variables meteorológicas a través de un sistema inalámbrico de adquisición de datos. <i>Revista de investigación, Desarrollo e Innovación</i> , 8(2), 333-341.	Desarrollar un sistema para el monitoreo inalámbrico de variables climáticas.	Software y hardware para un sistema de monitorización web de señales industriales.	Se utilizó un microcontrolador MSP432 encargado del muestreo y transmisión de las señales monitoreadas a una Raspberry PI	Integración de software y hardware que proporcionan datos en la web para su tratamiento y análisis.
Rivera, A. F. G., & Clavijo, F. V. (2016). Monitoreo De Sensores En Aplicaciones Web Embebidas. In <i>International Congress of Basic Sciences and Engineering CICI</i> (No. 105, pp. 4399-4402).	Desarrollar un prototipo de cámara multispectral para uso en agricultura de precisión.	Monitoreo de sensores; sistemas embebidos; servidor web; ordenador de placa única.	La Raspberry Pi 3 es una computadora de placa reducida. Procesador ARM Cortex-A53 de 64-bit	Monitoreo de sensores en tiempo real. La tecnología websocket facilita la interacción entre programas
Pérez, P. D. (2018). Sistema embebido basado en el monitoreo de consumo de energía eléctrica de electrodomésticos para el control del gasto. <i>Revista Tecnología Digital</i> . Vol. 8 (1), 69-80.	Monitorear el consumo de energía eléctrica y compararlo con las indicaciones de datos técnicos especificados y corroborar los resultados para localizar problemas de consumo excesivo.	Sensor de potencia, inducción electromagnética y sistema embebido.	Sensor de potencia T9B032 con arduino nano. ESP 8266 para enviar el monitoreo del sensor a una página web en tiempo real.	Un medidor de potencia, empleando el principio de inducción electromagnética (Bobinas), enviando la señal a un nano-arduino, y enviándolos vía web.

3.2. Sistemas espaciales de bajo costo

Existen diferentes tipos de sistemas de exploración espacial, los más comunes son los satélites, dependiendo de su aplicación, estos varían en tamaño y masa. Según [1, p. 5], los hay desde aquellos de varios metros de envergadura, como por ejemplo el Telescopio Espacial Hubble, hasta los de poco centímetros, como los empleados en los proyectos CanSat.

Los satélites miniaturizados son diseñados de la misma manera que los satélites grandes, sin embargo, dadas las limitaciones en cuanto a espacio y masa, sus misiones suelen ser mucho más específicas [1, p. 5]. Una ventaja importante es su bajo costo, corto tiempo de preparación y la simplicidad de sus diseños. A continuación, se define una tecnología aeroespacial de bajo costo muy popular, el cansat, así como las principales variables de medición.

Plataforma Cansat. El CanSat consiste en una plataforma que simula un sistema espacial, en este caso es un nanosatélite, el cual cabe en una lata de refresco de 350 ml. Los CanSat no orbitan, pero mediante su lanzamiento a través de un cohete (de hidropulsión, por ejemplo), un globo sonda o un avión a escala de control remoto, se pueden realizar pruebas y transmitir información de telemetría mientras descienden lentamente a tierra usando un paracaídas u otro método de descenso, desarrollando así la misión para la que fueron construidos [2, p. 3].

VARIABLES DE MEDICIÓN. A la tarea de fabricar y colocar en órbita satélites se le llama misión. Las misiones son bastante variadas, existen retransmisores para telecomunicaciones, telescopios ópticos, satélites con cámaras, etc. A pesar de esto y por muy diferentes que las misiones sean, siempre se cuenta con lo mismos subsistemas [1, p. 6]. Los subsistemas típicamente utilizan componentes comerciales para su electrónica. Dentro de estos subsistemas encontramos antenas, telemetría, sistema de potencia, control térmico, propulsión, entre otros. Las variables a medir depende de cada misión. Por ejemplo, un globo sonda “normalmente incluye una cámara, sensores de temperatura, sensores de presión, células solares, etc. y están conectados a equipos de seguimiento y comunicaciones. El globo se suelta y se eleva hasta que el globo estalla (normalmente a 25 km de altitud). En este punto las cargas útiles regresan a la superficie a través de paracaídas, transmitiendo datos de posición vía radio” [5, p. 2].

Estación terrena. La comunicación en tiempo real se lleva a cabo por una estación terrestre, esta forma parte del sistema de telemetría, rastreo y mando. “Una estación terrena es el equipo que se encuentra en algún punto de la Tierra y establece comunicación con el satélite; puede ser fija o estar instalada en lugares móviles, ya sean terrestres, marítimos o aéreos; son utilizadas para recibir, transmitir o ambas” [13, p. 363].

El tipo de equipo encontrado en una estación terrena varía según su uso (ya sea para comunicación transcontinental o un para un satélite) y depende también de la naturaleza de la misión, pero existen similitudes en todas las estaciones. Según [6, p. 625], los elementos principales dentro de una estación terrena son: Sistema de antena, Transmisión y recepción de equipos de radiofrecuencia, Equipos de telemetría, seguimiento y comando (TT&C), **Interfaz de usuario de datos**, recuperación de datos de la misión, centro de control de la estación y el reloj del sistema.

3.3. Ingeniería de software y aplicaciones web

La ingeniería de software, según [11], es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software, desde la etapas iniciales de la especificación del sistema hasta el mantenimiento de éste después de que se utiliza. Sin embargo, “aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser más efectivo en algunas circunstancias. El desarrollo informal es apropiado para el desarrollo de sistemas basados en web, los cuales requieren una mezcla de técnicas de software y de diseño gráfico” [11, p. 6].

De acuerdo con [7, p. 47], una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor y se define bajo tres niveles típicos: el nivel superior que interacciona con el usuario (el cliente web, normalmente un navegador), el nivel inferior que proporciona los datos (la base de datos) y el nivel intermedio que procesa los datos (el servidor web). Tanto el cliente como el

servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones.

En una aplicación web la interfaz gráfica está en el cliente web. “El cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP, explica. La parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador (JavaScript)” [7, p. 47]. Hoy los frameworks Frontend facilitan la creación de interfaces de usuario, ya que tienen una estructura de archivos y carpetas de código estandarizado (HTML, CSS, documentos JS, etc.), además de tener diseño responsivo.

3.4. Sistemas embebidos

Según [4, p. 2], existen varias definiciones de un sistema embebido o empotrado, aunque a grandes rasgos se puede entender como sistemas electrónicos de propósito específico programables. [4] nos da su propia definición: “Un sistema embebido es un sistema basado en microprocesador que está diseñado para controlar una función o rango de funciones y no está diseñado para ser programado por el usuario final de la misma manera que una PC”.

Así mismo, un sistema embebido es un sistema de información que cumple con el objetivo de la automatización de procesos. “Un sistema de información (SI) es un conjunto de elementos o componentes insurreccionados que recopilan (ingresan), manipulan (procesan), almacenan y difunden datos e información (de salida) y proporcionan una reacción correctiva (mecanismo de retroalimentación) para cumplir un objetivo” [12, p. 10], además de ser apoyo en el proceso de toma de decisiones.

- **Sensores:**El funcionamiento de un sistema embebido, en términos generales consta de: Entrada (sensores y/o periféricos), Proceso (Tiempo real), Salida (respuesta, resultados, periféricos). De este modo, a los sistemas embebidos se les puede conectar varios tipos de accesorios que le dan versatilidad en distintas tareas. Sus módulos de E/S analógicas y digitales suelen emplearse para digitalizar señales analógicas procedentes de sensores, de esta manera se tiene una interacción con diversos componentes para medir o controlar objetos del mundo real. La gran mayoría de estos son de bajo costo y bajo consumo de energía.
- **Microcontroladores:**El componente principal de un sistema embebido es el subsistema de procesamiento. “Los microcontroladores se pueden considerar como sistemas independientes con procesador, memoria y periféricos, de modo que en muchos casos todo lo que se necesita para usarlos dentro de un sistema integrado es agregar software” [4, p. 11]. Existen placas de desarrollo de hardware que facilitan el uso de la electrónica y programación de sistemas embebidos, ofreciendo bajo costo económico y facilidad de uso, siendo los más comunes: Arduino, Raspberry y PIC. Su principal ventaja es tener una gran comunidad de usuarios que lo respaldan. Por lo que hay disponibles multitud de tutoriales, herramientas, ejemplos de uso, librerías y hardware de E/S.

- Comunicaciones: Para comunicar un sistema embebido existen soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos, en la tabla 2 vemos las prestaciones que ofrecen módulos de transmisión inalámbrica más usados.

Tabla 2. Comparacion entre sistemas de comunicación.

	Wi-Fi	Bluethooth	XBEE
Velocidad	≤ 50 Mbps	1 Mbps	(máx) :250.000b/s
Rango de alcance	50 metros	10 metros	1200 metros (al aire libre)
Precio	Caro	Medio	Barato
Configuración	Compleja	Compleja	Simple
Duración batería	Horas	Días	Años

Fuente: Elaboración propia.

Como se observa los módulos XBee son una tecnología que hace frente a las necesidades de bajo costo y redes de bajo consumo, con un estándar global abierto.

4. Materiales y métodos

En esta sección se presenta los componente de la plataforma embebida, así como el diseño de interfaz de usuario mediante el uso de tecnologías web.

4.1. Subsistema de sensores

La elección de los sensores se realizó en función de la misión para el cohe-te hidropulsado, es decir, deberá tener módulos de censado que midan las siguientes variables: presión, temperatura, humedad critica longitud y latitud mediante GPS, aceleración en tres ejes y altitud.

Por lo tanto se obtuvieron los siguientes sensores:

Tabla 3. Modelos de sensores utilizados.

Variable	Modelo del sensor
Temperatura y humedad	DHT11
Aceleración, Altura, presión barométrica, magnetómetro	IMU-GY80
GPS	GP-635T
Sensor de polvo	GP2Y1010

Cabe mencionar que se incluyó un sensor de calidad del aire, en específico un sensor de polvo. A continuación se hace un análisis de criterio de selección de cada uno de los sensores adquiridos.

1. DHT11. Es un sensor digital de temperatura y humedad relativa, un dispositivo sencillo, de bajo costo y de fácil uso. A nivel de software se dispone de librerías para Arduino. En cuanto al hardware, solo es necesario conectar el pin VCC de alimentación a 3-5V, el pin GND a Tierra (0V) y el pin de datos a un pin digital en un Arduino.
2. IMU-GY80. Es un módulo muy práctico pues incluye un giroscopio, acelerómetro, brújula digital, y un sensor de presión barométrica, todo en un mismo chip. Todos los sensores individuales son accesibles a través de I2C lo que sólo necesita 4 conexiones para acceder a todos los sensores. Lo que nos ayuda a ahorrar mucho espacio.
3. GP-735T. Este GPS es delgado y tiene una antena a interna, lo que es importante pues no se tiene mucho espacio para trabajar, además tiene una sensibilidad de seguimiento muy buena y un arranque en frío de unos 27 segundos aproximadamente. Su voltaje de operación entre 3,3 V a 5,5 V. Lo que garantiza un buen rendimiento de batería.
4. GP2Y1010. Permite medir la concentración de polvo en el aire usando la luz infrarroja reflejada, incluso partículas humo muy finas. Tiene un consumo de corriente muy bajo (máximo 20mA, valor típico 11mA), además nos dará un indicador de la calidad del aire.

Elección de dispositivos de manejo de datos (Selección de antena).

En la siguiente tabla vemos las prestaciones que ofrecen módulos de transmisión inalámbrica más usados.

Tabla 4. Módulos de transmisión inalámbrica más usados.

	Wi-Fi	Bluetooth	XBEE S2C
Velocidad	<50 Mbps	1 Mbps	(máx) :250.000b/s
Rango de alcance	50 metros	10 metros	1200 metros (al aire libre)
Precio	Caro	Medio	Barato
Configuración	Compleja	Compleja	Simple
Duración batería	Horas	Días	Años

Mientras la tecnología Bluetooth está orientada a la movilidad del usuario y a eliminar el cableado entre los dispositivos de corta distancia, Xbee está orientado hacia el control remoto y la automatización, además el alcance en la transmisión es bastante mayor. Cabe decir, que los factores de programación de cada antena no condicionan la elección. Solo se considera nuestra prioridad un alcance amplio en el envío de datos, por tanto se decanta por el XBee S2C por ser un módulo de bajo costo y bajo consumo de energía. Con él se tiene un alcance de 1200 metros al aire libre. Esta optimizado la conexión con microcontroladores lo que ayuda con el objetivo del proyecto e incorpora la comunicación SPI (serial

peripheral Interface) que provee un intercambio de datos con una alta velocidad entre dispositivos.

4.2. Subsistema de potencia y diseño electrónico

Circuito eléctrico. Al igual que con los sensores se hace un pequeño análisis de criterio de selección del sistema principal (Arduino) y subsistema de potencia.

1. Arduino Pro mini. El chip Atmega328P-AU corriendo a 16Mhz es una de las grandes ventajas de esta placa, ya que permite mantener un procesamiento óptimo a bajo consumo de energía. Se eligió por su bajo costo y diseño, ya que está pensado para aplicaciones que requieren el menor tamaño posible, además de su programación en un lenguaje de alto nivel: Processing que es similar a C++.
2. XL6009E1. Este módulo de fuente de alimentación actúa como regulador-elevador de voltaje, tiene un rango que va de 3V hasta 35V, es de alto rendimiento en términos de corriente de salida, proporcionando una corriente máxima de 4 amperes (con disipación de calor necesaria), es usado para aplicaciones en las que se necesita un nivel de tensión constante a la salida ante cualquier tipo de variaciones de voltaje a la entrada del módulo. De este modo, podemos conectar varios dispositivos sin una caída de voltaje.

Distribución de potencia. En la tabla 5 se presentan las características eléctricas básicas de los circuitos que conforman el sistema eléctrico:

Tabla 5. Características eléctricas básicas de los componentes de censado.

Arduino	<ul style="list-style-type: none"> • Voltaje de funcionamiento: 5v • Voltaje de entrada: 5 – (12v, pin "RAW") • Corriente máxima: 150mA
Xbee S2C	<ul style="list-style-type: none"> • Voltaje de funcionamiento: 2.7V a 3.6V • potencia Transmisión: 6.3mW output (+8dBm) • Corriente – Recepción: 31mA • Bajo consumo de corriente (1uA en modo sleep)
XL6009E1	<ul style="list-style-type: none"> • Voltaje de entrada: 3V-32V (voltaje óptimo 5V-32V) • Voltaje de salida: 5-35V Ajustable • Intensidad máxima de entrada: 4A (usar disipador térmico) • Intensidad máxima de salida: 3A • Eficiencia aproximada: 94% • Frecuencia de trabajo: 400KHz • Rango de temperatura trabajo: -40 a +85 grados centígrados
DHT11	<ul style="list-style-type: none"> • Voltaje de operación: 3.3 o 5V DC • Corrientes de operación: (VCC=5V, T=25°C) → 0.5 a 2.5 mA
GP2Y1010AU0F	<ul style="list-style-type: none"> • Voltaje de funcionamiento típico: 4.5V a 5.5V (MAX: 7V) • Bajo consumo de corriente (MAX: 20mA)
GP-735	<ul style="list-style-type: none"> • Corriente de Recepción: 37mA • Voltaje de entrada: 3.1V ~ 5.5V
IMU GY-80	<ul style="list-style-type: none"> • Voltaje: 3.3V a 5V • Derivado de la especificación para el bus I2C. Arduino El máximo absoluto para cualquier pin IO individual es de 40 mA, por tanto sería el consumo para el bus I2C.
Batería	<ul style="list-style-type: none"> • Batería de li-tio-polímero (LiPo) de 1200mAh, 3.7v 1C (1 celda), con conector tipo JST hembra.

Utilizando la ley de ohm se explica porque el subsistema de potencia es capaz de alimentar todos los otros subsistemas, en la figura 1 se muestran las fórmulas para la demostración:

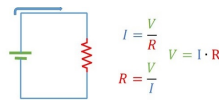


Fig. 1. Formulación matemática de la ley de ohm.
Fuente: *Elaboración propia.*

Las relaciones básicas en el subsistema de censado pueden verse como un circuito equivalente conectado en paralelo (figura 2). Donde Sn presentan los sensores.

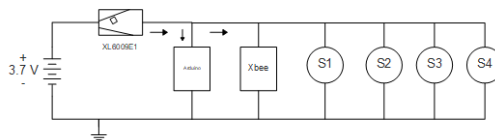


Fig. 2. Circuito equivalente conectado en paralelo.
Fuente: *Elaboración propia.*

El circuito requiere un voltaje mínimo (lógico) de 3.3v para funcionar correctamente, sin embargo las diferentes cargas útiles pueden requerir un voltaje superior. Para cubrir esa necesidad se incluyó un voltaje secundario con regulador conmutado (dispositivo XL6009E1). Por lo tanto, el circuito se puede redefinir con una fuente de alimentación, fijando la tensión en 5 voltios, quedando el circuito anterior como el que se muestra en la figura 3:

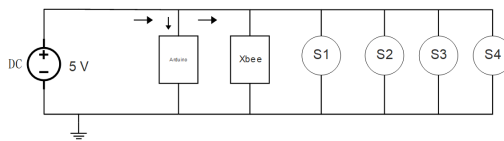


Fig. 3. Reducción del circuito equivalente conectado en paralelo.
Fuente: *Elaboración propia.*

Como se conocen los voltajes y corrientes correspondientes a cada dispositivo y teniendo en cuenta que el circuito esta en paralelo, este se puede simplificar

utilizando el concepto de la resistencia equivalente. Se utilizará la formula de la figura 4, para calcular la resistencia total.

$$R_t = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots}$$

Fig. 4. Formula de resistencia total.

Antes de proceder con la operación, se calculan las resistencias que se tienen en cada dispositivo dentro del subsistema:

$$R = \frac{V}{I}$$

$$\text{Arduino} = \frac{5v}{0.150A} = \frac{100}{30} \Omega$$

$$\text{Xbee} = \frac{5v}{0.031A} = \frac{5000}{31} \Omega$$

$$\text{DHT11} = \frac{5v}{0.0025A} = 2000 \Omega$$

$$\text{GP2Y1010AU0F} = \frac{5v}{0.020A} = 250 \Omega$$

$$\text{GPS} = \frac{5v}{0.037A} = \frac{5000}{37} \Omega$$

$$\text{GY80} = \frac{5v}{0.040A} = 125 \Omega$$

Fig. 5. Calculo de las resistencias del subsistema.

Teniendo el valor de las resistencias, podemos calcular la resistencia equivalente, entonces:

$$R_t = \frac{1}{\frac{3}{100} + \frac{31}{5000} + \frac{1}{2000} + \frac{1}{25} + \frac{37}{5000} + \frac{1}{25}} \Omega$$

$$R_t = \frac{10000}{561} \Omega \approx 17.825311943$$

Fig. 6. Resistencia equivalente.

Con estas operaciones ya es posible calcular un nuevo circuito con la ley de Ohm. En comparación con el circuito en serie la cantidad de resistencia es menor porque se divide la corriente. De esta forma, en la figura 7, tenemos el circuito equivalente.

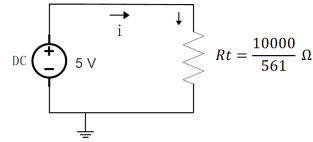


Fig. 7. Reducción del circuito equivalente conectado en paralelo.
Fuente: Elaboración propia.

Con el circuito equivalente se obtiene la corriente total del circuito, usando la Ley de Ohm. Tenemos el voltaje (5V), la resistencia total (R_t), entonces la corriente total se obtiene como en la figura 8.

$$I_t = \frac{V}{R} = \frac{5v}{\frac{10000}{561}} = \frac{2805}{10000} = 280.5 \text{ mA}$$

Fig. 8. Corriente total.

La batería que se está utilizando nos da una corriente de 1200 mAh, con lo cual podemos calcular el tiempo de descarga de la batería en función al subsistema de sensado, entonces tenemos la resistencia equivalente (Fig. 9).

$$\frac{280.5mA}{1200mAh} \approx 0.23 \text{ hrs}$$

Si tenemos que: $\frac{1 \text{ h}}{0.23 \text{ h}} = \frac{60 \text{ min}}{t}$

Entonces tenemos un tiempo aproximado de

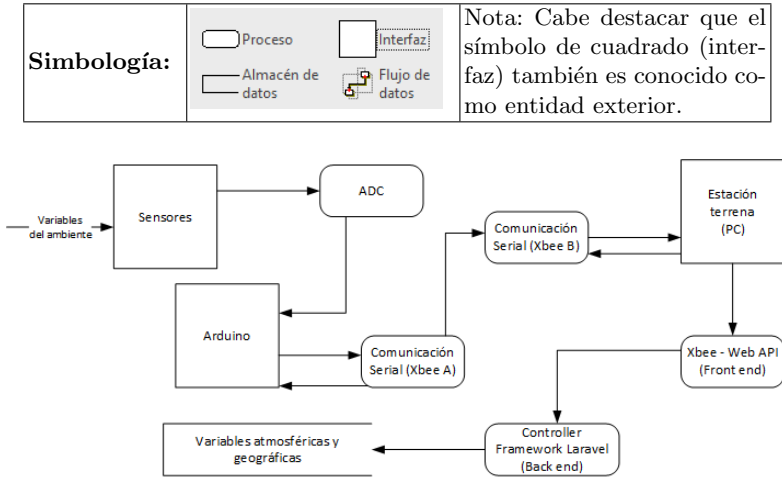
$$t = 13.8 \text{ min}$$

Fig. 9. Resistencia equivalente.

Con lo cual el subsistema que alimentamos con la batería tendría una corriente apropiadamente suministrada por un total de 14 min aproximados, sin sufrir pérdidas de energía, con lo cual para al objetivo que persigue el cohete hidropulsado que es la medición de variables después de su lanzamiento, es cubierto por el sistema de potencia pues se satisfacen las necesidades de energía.

4.3. Subsistema de comunicaciones y manejo de datos

Diagrama de flujo de datos. Se muestra la representación gráfica del flujo de datos a través de un diagrama de flujo de datos o DFD (sus siglas en español e inglés) para el sistema de información que se encargara del manejo de datos y la comunicación entre el cohete y la estación terrena.

 Almacén de datos

Flujo de datos

Fig. 10. Diagrama de Contexto: Nivel 0.
Fuente: *Elaboración propia.*

La figura 10 nos muestra las variables del entorno o ambiente que serán el insumo de entrada para nuestro sistema. Entre los sensores y el sistema Arduino existe un proceso de conversión de señal analógica a digital. Posteriormente se realiza el proceso de comunicación inalámbrica por medio de dos sistemas Xbee (A y B). Donde el Xbee A está transmitiendo los datos que se tomaron de los sensores y el Xbee B los recibe y transfiere a la computadora. La computadora actúa como un servidor remoto y se comunica con el Xbee B mediante un API en JavaScript. Teniendo los datos del lado del cliente (front end) enviándonos la información para su almacenamiento, por medio de un proceso de guardado que se realiza por un framework (back end).

4.4. Diagrama de comunicaciones

A grandes rasgos la comunicación es la transferencia de información con sentido desde un lugar (remitente, origen, fuente, transmisor) a otro lugar (destino, receptor), tal y como se muestra en la figura 11.

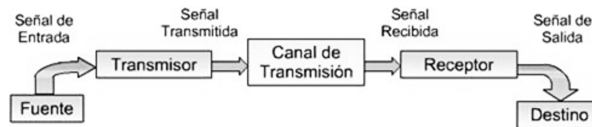


Fig. 11. Elementos básicos de un sistema de comunicaciones.
Fuente: *Elaboración propia.*

Bajo la premisa anterior, el sistema de telemetría, para la misión del cohete hidropropulsado, lleva a cabo la implementación de este concepto como se muestra en la figura 12. Aquí se presentan bloques del sistema de comunicación, donde la fuente de información principal proviene del exterior y el transductor de entrada serán las señales de los sensores de entrada.

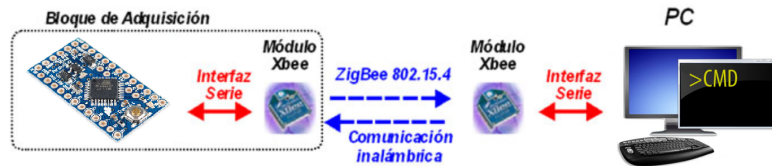


Fig. 12. Elementos básicos de un sistema de comunicaciones.

Fuente: Elaboración propia.

En este punto se puede describir la comunicación que tendrán entre cada sistema principal de la misión.

1. Concepción General del Sistema de Adquisición de Datos (Bloque de Adquisición): El Bloque de Adquisición, ubicado junto al grupo de sensores, está integrado por un microcontrolador y conectado con un módulo de comunicación inalámbrica XBee. El sistema podrá adquirir varias señales analógicas. La presencia de un microcontrolador con entradas analógicas permite digitalizar la señal analógica recibida, lo que facilitará su procesamiento y transmisión. El usuario recibirá la información del monitoreo en tiempo real o la podrá consultar cuando desee, dado que los datos adquiridos se almacenarán.
2. Comunicación inalámbrica: Los XBee está basados en el estándar ZigBee. ZigBee es una tecnología inalámbrica basada en el protocolo IEEE 802.15.4, una tecnología de red inalámbrica que opera a 2,4GHz. Desde el programa que se ejecuta en el microcontrolador se configura el módulo conectado a él, y desde la PC se configura el otro módulo con la herramienta XCTU conectado a esta. El usuario solo tiene que garantizar que el dato a transmitir esté en la salida del puerto serie del microcontrolador. Los módulos XBee se encargan de ejecutar todas las tareas necesarias para la comunicación.
3. Comunicación con el servidor: Se trabaja con sockets en Node.js, con esto tenemos un API (front end) que se utiliza para comunicarse con el módulo de radio de protocolo 802.15.4 y ZigBee. El socket no se encarga de la conexión serie en sí, pero se podrá conectar a módulos como serialport (una dependencia npm), que facilitan el acceso al puerto serial. Por otra parte, para lograr una integración entre el xbee y el servidor (en back end) se tienen dos sitios web separados capaces de comunicarse a través de un servicio web. Se tiene el primer sitio web (A) que interactúa con los datos directamente, es decir, el módulo xbee y este expone la API, para que el otro sitio web (B), bajo el framework Laravel, lo consuma. De este modo el framework

podrá interactuar con la base datos (en un futuro) y crear los registros de los datos capturados.

4.5. Protocolo de recepción y muestreo de datos

En la figura 13 muestra un protocolo de recepción y muestreo de datos básico y se establece de la siguiente manera:



Fig. 13. El protocolo de recepción y muestreo de datos.

Fuente: Elaboración propia.

- **Recepción de datos.** Este objetivo se cumple gracias al módulo hardware XBee, que captura señales y las pone de forma digital. Los módulos XBee tienen dos formas de trabajo: el modo transparente y el modo API. Para el proyecto se utilizó el modo transparente. Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control.
- **Muestreo de datos.** El objetivo es analizar las señales que se obtuvieron en la etapa anterior. Como se mencionó la comunicación es transparente, es decir, se recibe lo que se envía. En el modo transparente el dispositivo funciona como un puente Serial-Zigbee, es decir, la información que se recibe por la USART es transmitida por el aire, en una frecuencia de trabajo de 433MHz, debido a que es una banda de libre uso.
- **Presentación web de los datos muestreados.** El objetivo de esta etapa es la presentación web de los datos muestreados. Este objetivo se cumplió diseñando dos servidores web que usan JavaScript y node.js.

4.6. Diagrama de bloques de estación terrena

A continuación, en la figura 14 se presenta el diagrama de bloques para el software de la estación terrena:

En primer lugar se tiene al módulo Xbee, este es el encargado de recibir los datos que provienen de los sensores. Este se conecta a la PC haciendo uso de Serialport-API. Se usa los sockets de JavaScript para lograr un enlace UART¹. Se puede decir que se tiene un primer cliente que le hará peticiones a un servidor web.

¹ **UART**, son las siglas en inglés de Universal Asynchronous Receiver-Transmitter, en español: Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie.

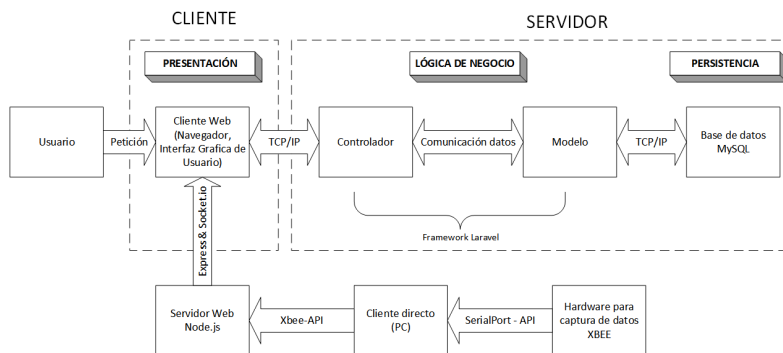


Fig. 14. Diagrama de bloques para el software de estación terrena.
 Fuente: *Elaboración propia.*

Una vez establecido el enlace UART, es preciso enviar estos datos al servidor web, el cual está alojado en una máquina remota (PC). Para lograr esto se hace uso de Node.js y express.js este último es un framework web. El módulo Express, es el encargado de enviar el archivo HTML al cliente (el front end), cuando este realice solicitudes al portal, y así el módulo xbee-api, se dedicara a manejar el tráfico de información (las variables a monitorear) entre el cliente y el servidor.

La arquitectura utilizada para el diseño de la aplicación (el back end) está dividida en 3 capas (Presentación, Lógica del Negocio y Datos) y 2 niveles (Cliente y Servidor). La capa de presentación posee la interfaz gráfica de usuario. La capa de lógica del negocio posee el controlador. Finalmente, la capa de datos o de persistencia almacena la información referente a los datos del negocio (las variables de los sensores). Para tener una arquitectura y diseño solido se usa un marco de trabajo o framework llamado Laravel basado en MVC. Por último, la persistencia se puede obtener por medio de una base de datos estructurada entre Laravel y MySQL, por ejemplo.

4.7. Diseño gráfico y funcional

Al ser una aplicación web, la funcionalidad de la interfaz deriva de su facilidad de consultar información mediante un navegador web (en smartphones, tablet, portátil o PC) conectado al servidor, sin tener que instalar el software (lo que ocurriría con un software de escritorio). Para construir la interfaz gráfica se utilizó Bootstrap, este framework nos proporciona un "cascaron" con un diseño profesional y editable, sin tener que lidiar con el diseño de un sitio, sobre el cual se construyó la interfaz web de la estación terrena, además ofrece la ventaja del ahorro de tiempo para integrar la información ya que este cuenta con una estructura definida y esta optimizado para ser responsivo.

5. Resultados y conclusiones

Se cumplió con los objetivos, obteniéndose un sistema integral de bajo costo, que además ofrece la ventaja de ser un sistema con portabilidad y al mismo tiempo escalable. Este software deja un precedente de como hacer software de estaciones terrenas para este tipo de concursos. Se obtiene un circuito encargado de transmitir información de las diferentes variables que se necesitaban cubrir, con un peso 165 gramos de carga total. En la figura 15 se presenta el circuito encargado de la telemetría.

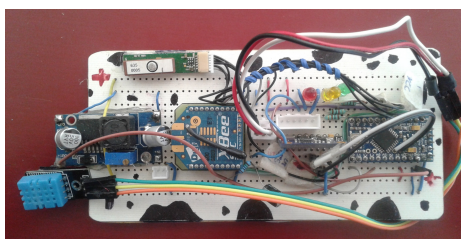


Fig. 15. Circuito de telemetría.

Asimismo, durante el concurso el proceso de medición funcionó correctamente. Aunado a esto el software fue la mejor estación terrena. Lo que en consecuencia se logrará tener un primer lugar en el concurso. Por otra parte, La interfaz del sistema de monitoreo, permitió la manipulación de la estación por parte de otros usuarios de manera sencilla, a la vez que permitió la comunicación con computadoras, por ejemplo en el caso de la geolocalización, en el cual los jueces podían verificar, en tiempo real, la ubicación a través del API embebido de Google Maps.

Finalmente, los resultados permiten concluir que es posible desarrollar estaciones de monitoreo o estaciones terrenas de bajo costo, a partir de dispositivos electrónicos y software de uso común, con rendimiento similar al de los equipos comerciales. Sin embargo, hay que ser conscientes de que este tipo de desarrollos depende siempre de variables como: internet, dominio o ambiente, infraestructura, entre otras cosas. A pesar de esto, se puede decir que la estación terrena es un sistema integral que podría reducir considerablemente los costos en un proyecto real en el tema de la tecnología espacial o de meteorología, por ejemplo.

En la figura 16 se muestra la vista del diseño gráfico de la interfaz para la estación terrena.

Agradecimientos. Este trabajo ha sido apoyado por el Instituto Politécnico Nacional, México (IPN) y el Centro de Desarrollo Aeroespacial (CDA) del IPN. Además este trabajo es desarrollado con recursos e instalaciones del IPN. Cabe resaltar que los resultados de este trabajo fueron realizados por el tesista de Nivel

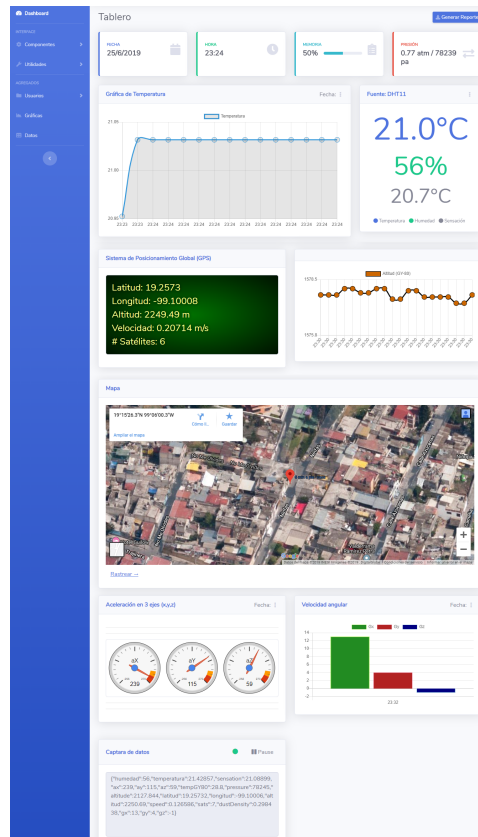


Fig. 16. Vista de la interfaz terrena.

Maestría Alberto Carlos Pérez Cadena. También, se le agradece a los revisores que aportaron sus valiosos conocimientos para mejora del presente artículo.

Referencias

1. Estructura y control térmico de nanosatélites (Diciembre 2013)
2. Agencia Espacial Mexicana: Ingeniería de sistemas espaciales: Aplicado a una misión cansat. https://www.educacionespacial.aem.gob.mx/images_ise/pdf/01introduccion.pdf/ (2017), consultado el 22-04-2019
3. Bertoglio, O.J., Johansen, O.: Introducción a la teoría general de sistemas. Editorial limusa (1982)
4. Heath, S.: Embedded systems design. Elsevier (2002)
5. LaCombe, J.C., Wang, E.L., Nicolescu, M., Rivera, P., Poe, B.: Design experiences with a student satellite program. In: Proceedings of the 2007 American Society for Engineering Education Pacific Southwest Annual Conference (2007)

6. Larson, W.J., Wertz, J.R.: Space mission analysis and design. Tech. rep., Torrance, CA (United States); Microcosm, Inc. (1992)
7. Luján-Mora, S.: Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario (2002)
8. Pérez, P.D.: Sistema embebido basado en el monitoreo de consumo de energía eléctrica de electrodomésticos para el control del gasto. *Revista Tecnología Digital*. Vol 8(1), 69–80 (2018)
9. Rivera, A.F.G., Clavijo, F.V.: Monitoreo de sensores en aplicaciones web embebidas. In: *International Congress of Basic Sciences and Engineering CICI*. pp. 4399–4402. No. 105 (2016)
10. Ruiz-Ayala, D.C., Vides-Herrera, C.A., Pardo-García, A.: Monitoreo de variables meteorológicas a través de un sistema inalámbrico de adquisición de datos. *Revista de investigación, Desarrollo e Innovación* 8(2), 333–341 (2018)
11. Sommerville, I., Galipienso, M.I.A.: *Ingeniería del software*. Pearson Educación (2005)
12. Stair, R., Moisiadis, F., Genrich, R., Reynolds, G.: *Principles of information systems: A managerial approach* (2010)
13. Vela, R.N.: *Comunicaciones por Satélite*. México. International Thomson Editores (2003)

Teaching Basic Concepts: Geometric Forms and Colors on a NAO Robot Platform

Diana Olvera, Uriel Escalona, Humberto Sossa

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, Mexico
a150815@sagitario.cic.ipn.mx

Abstract. Basic geometry figures and colors are some of the first approach on children mathematics learning, this process tend to be long-time and often is needed several sessions to learn these concepts correctly. Classical teaching models require a teacher and students, where teacher show the concept (name, form) and student repeat. Cases where the concept is not learned in a group, it is required personal sessions one-to-one, taking more time that teacher can give to each student. In this paper we propose the use of NAO robot as an educative platform to teach colors and geometry forms to children, using voice and image processing, with a game-like structure ensuring the learning process result in correct learning.

Keywords: NAO robot, teaching robots, geometric forms.

1 Introduction

The use of Information and Communications Technologies (ICT) on daily life is a reality of our times, in education is a complement that allow a better transmission and demonstration of knowledge, however, this most be implemented with the correct planning over academic activities [3].

Education robotic offer a safe environment where kids can test a hypothesis and problem solutions according to age always supported by learning process well defined and tested to not engrave or threatened the development of kids [14]. Initial experiments as [2, 11, 8] show the benefits obtained when a computational systems helps in the education, emphasizing that the robot is not a replace of humans, only is used as a reinforcement.

This paper focuses on the idea of providing a robotic tool that can interact with children as many times as necessary, this being the main advantage of the use of robotic systems, since these do not show fatigue or boredom, so they can continue working the time that is necessary until the concepts to work are learned correctly.

1.1 Educational Robotic

The Educational Robotic (ER), implemented through a pedagogical approach, fosters a learning environment in which the use of robots in the classroom is justified to promote the learning of different areas [19, 2, 12].

It is defined in 1989 by Vivet as: "Activity of conception, creation and implementation operation, for didactic purposes, of technological objects, which are reduced reproductions very faithful and meaningful processes and robotic tools that are used daily, and that they are becoming more common in our social, productive and cultural environment ", or, as the discipline that allows to conceive, design and develop educational robots, to promote that the students begin in science and technology from an early age, in order to explore from another perspective the interaction with a robot to favor cognitive processes [13].

The six main reported areas that have been proposed in educational robotics are:

1. Support in primary and secondary education.
2. Adults in professional training.
3. Robotics applied to people with disabilities.
4. Pedagogical robotics to facilitate the development of cognitive processes and representation.
5. Robotics as a tool in laboratories.

The application of this discipline aims to explore how attractive it is for learners, as well as for educators the idea of learning by playing. With the use of educational robotics, technological approaches can be made from an early age, and breaking with the myths that this is only a science fiction thing.

In order to verify the objectives of educational robotics as an integrating discipline of different areas of knowledge, the development of two individual processes is necessary but highly dependent. On the one hand, functions must be established from the point of view of engineering for the study and process of conceiving, designing and constructing robotic mechanisms; and a second function, from the didactic point of view, to verify that these mechanisms effectively meet the educational purposes for which they were developed, which involves research in the disciplines of knowledge of education, teaching and learning. Therefore, this paper work focuses on providing educational skills to a robotic system that meets the basic requirements for execution in an educational environment.

2 Methodology and NAO Robot

2.1 NAO Robot

NAO robot (Fig. 1) is part of humanoid robots, referring to an artificial entity which has a similar shape to humans. Been one of the first humanoid robot released in 2006 by Aldebaran Company in France. In table 1 the main physical characteristics are shown.



Fig. 1. Humanoid Nao robot.

Table 1. NAO robot main characteristics.

Feature	Value
Size	573x311x275 mm
Weight	5.4 Kg
Processor	ATOM z530 1.6 GHz
RAM	1 Gb
Hard disk	8 Gb

With 25 grades of freedom, 2 HD cameras, touch sensors, 4 omnidirectional microphones and 2 speakers positioned around the body this robot can interact with the environment similar as a human can do.

The cameras are positioned in the head were proportionate a wide angle as shown in Fig 2.

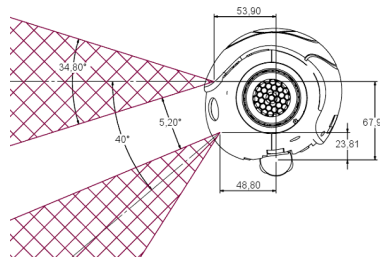


Fig. 2. Position and angle view of NAO cameras.

It can be programmed in several software as NAOQI-OS, Webots and Choregraphe, using C++, Java, Java Script and Python as programming languages.

NAO Programming Choregraphe (Fig. 3) is an intuitive graphic environment where algorithms are built by series of blocks, predefined sets allow the robot to complete from easy to complex tasks as face recognition. These blocks follow a sequence on time to execute the algorithm created in Python.

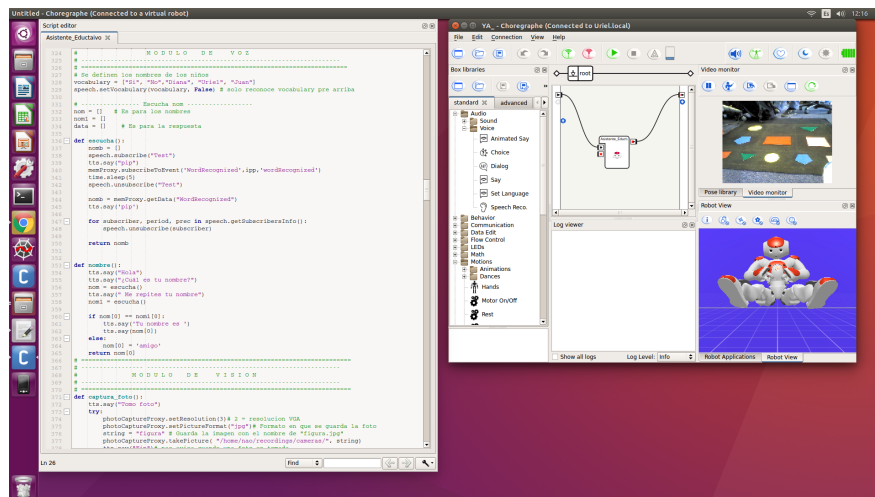


Fig. 3. Choregraphe, software to program NAO behaviour in block by Python.

2.2 Educational Environment with NAO Robot Setup

Among the first knowledge acquired in childhood are colors and figures, these are considered the first steps to the mathematical world and are essential to recognize the environment in which we find ourselves.

Color Selection The teaching of colors is a main area for the exploration of the environment, which is taught in the first years of the child's life. The teaching of colors for children has no age limits nor is there a metric of how many colors they should learn at an early age. However, it is suggested that children start with primary colors and then secondary colors, or, make a mixture of them, preferably 5 to 6 vibrant colors such as red, pink, green, blue, can help strengthen their learning.

According to [15] children can learn color in three steps:

1. Color association: First step is to hear name color, this can be made by the teacher or in case of ER by a robot.
2. Point out a color: Second step help children to know the differences between colors, in this activity children touch different colors and teacher or robot correct the name or reward a correct choice.

3. Name color: As a final step the children must be able to name the colors without the need to point out, this step make clear that children have learn correctly the colors.

When it comes to learning colors at an early age, patience is essential and although in some children the process may be short in other cases it can be a long process. So if the child already knows the fundamentals and differences, this does not mean that they should stop, but it is time to start teaching tertiary colors, as, a blue color has color variants for example color navy blue, sky blue, among others.

Geometry Figures Selection The initial understanding of geometry emerges in the early years as a physical knowledge of space [17]; it is possible to create activities to increase the learning of the geometry, the common geometric figures (Fig. 4) at an early age to use are flat (2D) include the circle, the triangle, the square, the rectangle, the pentagon, the rhombus and the ellipse, and the spatial figures (3D) include the sphere, the cylinder, the cone, the cube and the rectangular prism (Fig. 5).

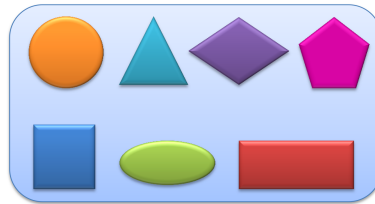


Fig. 4. 2D geometry figures.

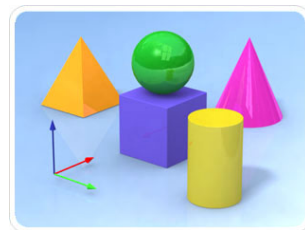


Fig. 5. 3D spatial figures.

It is known that children do not have the appropriate language to define the geometric figures with the appropriate word, however, children are able to find the similarities and differences in the forms present in the environment that

surrounds them. The goal for these ages is precisely to develop this ability to discriminate one way and another, since they invent their own reference points using their everyday experiences.

Pattern Recognition To carry out the recognition of patterns there are many techniques of vision by computer [16], it is necessary to provide the robot with this type of characteristics to allow a mobility and interpretation to develop in the world around it. The accuracy of these algorithms depends on two major aspects. The first is the limitation of the hardware itself and the other is the robustness of the program, where the ideal is that there is a minimum number of failures [20] to be carried in real time [5, 4].

In this paper we propose the used of Ramer Douglas Peucker (RDP) algorithm, created in 1970 [6], it is an algorithm used to simplify and reduce the number of points on a line or curve, is commonly used in geographic map processing, or in the recognition system of certain objects [9].

The RDP algorithm (Fig. 6) is explained in [10, 7, 18, 1]. As follow, the points within a sector of points are located and renamed assuming an imaginary line between the first and last points in the system of the points of the curve, determining the furthest point for this sector as the first and the last point of the endpoints. In spite of this, if this point or all other points between the start and end points are closer than a fixed distance, eliminate all those points between the two points. However, in case the imaginary line is greater than the value "Standard of Tolerance or Disparity (e)", it means that these points have already been included within the points of the curve [1].

3 Learning Process and Results

A board with size 60 cm by 80 cm containing geometric figures of different colors is developed, besides a replica of these is added that will help the NAO robot with the interaction as shown in Fig 7.

This board include a blue pentagon, green triangle, pink square, yellow circle and red rectangle. All figures can be changed of position as well as rotated in 5 different spaces, the purpose is to change the position of the figures in each session to secure that children learn the object and not the space.

The main idea is that the child can interact with the geometry figures and color with different senses as sight and touch and NAO robot helps with sounds.

Teaching color with NAO robot follows this steps:

1. The NAO robot introduces itself and asks the name of the children,
 - (a) The name is recorded for future interaction by microphone and natural language process algorithms,
2. NAO robot points out the different colors that exist on the board in order from left to right, indicating by sound the name of each one:
 - (a) a photograph of the board is taken by the front camera of the NAO robot to know the position of each color using the RDP algorithm.

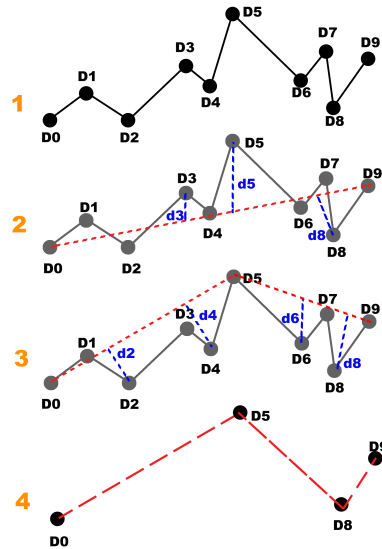


Fig. 6. RDP algorithm finds the relevant points of a series of points using a tolerance distance (e) as a limit value.

3. NAO robot asks the children to point and touch one of the 5 color:
 - (a) a photograph of the board is taken with the hand of the children to identify which color has been hindered,
 - (b) If the selection is correct, the NAO robot congratulates the children with his name and is instructed to take a prize,
 - (c) If the selection is incorrect, the NAO robot corrects the children indicating the correct answer.
4. The process in (3) is repeated up to a maximum of 5 times per session.

In the case of geometrical figures the process is very similar done through the following steps:

1. The NAO robot introduces itself and asks the name of the children,
 - (a) The name is recorded for future interaction by microphones and natural language process algorithms.
2. NAO robot points out the different geometrical figures that exist on the board in order from left to right, indicating by sound the name of each one:
 - (a) a photograph of the board is taken by the front camera of the NAO robot to know the position of each geometric figure using the RDP algorithm.
3. NAO robot asks the children to point and touch one of the 5 figures randomly:
 - (a) a photograph of the board is taken with the hand of the children to identify which figure has been hindered.

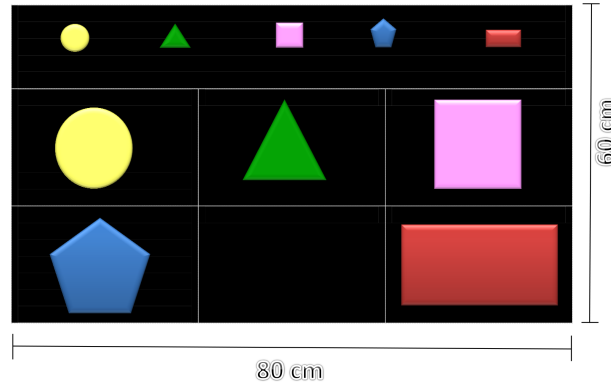


Fig. 7. Board with geometric figures and replicas for NAO robot.

- (b) If the selection is correct, the NAO robot congratulates the children with his name and is instructed to take a prize.
 - (c) If the selection is incorrect, the NAO robot corrects the children indicating the correct answer.
4. The process in (3) is repeated up to a maximum of 5 times per session.

In both cases, color learning and geometric figure learning NAO robot uses the front camera to take a photograph of the interactive board to know the position of each figure (Fig. 8).

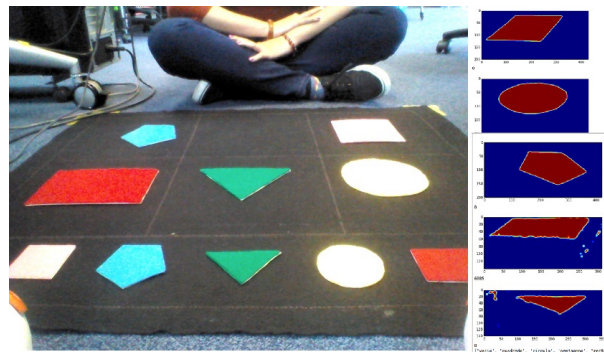


Fig. 8. Photograph from NAO robot point of view and RPD algorithm process.

As light conditions usually can not be controlled in a classroom the RGB information from the images is not enough to determine the figure in each section, so RDP algorithm provides a better solution as follows:

- Image taken from the frontal camera is divided in 5 sections which contain the figures.

- Images are binarized with a threshold near to 0 to separate black color of background from images.
- Blur method eliminates small noises as salt and pepper.
- RDP algorithm detects the number of edges in each image as follow:
 - square and rectangle result in 4 edges,
 - Triangle result in 3 edges,
 - Pentagon result in 5 edges,
 - Circle result in more than 7 edges.
- In the case of square and rectangle as both have 4 edges, internal value is taken, since pink is lighter than red this helps to correctly classified each figure.

Four children were selected with ages between 3 and 4 years to interact with NAO robot, at the beginning these children did not know correctly the colors and figures of the board. A series of 30 sessions were carried out interspersing the teaching of colors and geometrical figures. Their learning of these geometrical figures and color are shown in Fig 9.

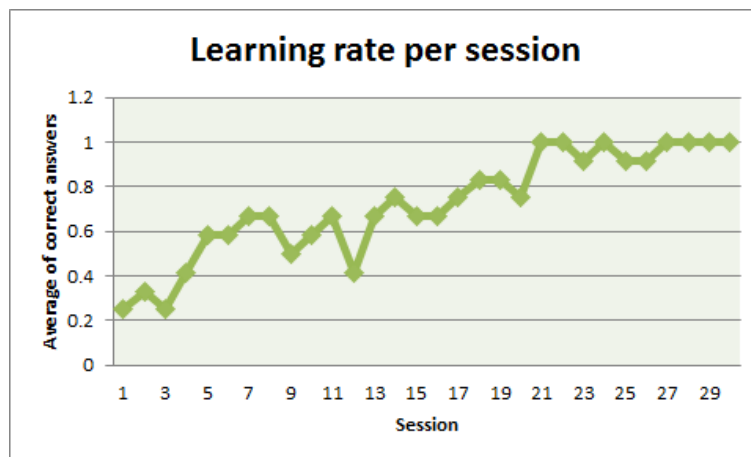


Fig. 9. Learning rate of geometrical figures and color by children.

It is observed that at the beginning of the sessions, the children could not correctly identify the geometrical shapes or colors, being their random choice, so that their percentage of success is close to 25%, as they progressed in their learning the children began to select correctly, and at the end in the last 4 sessions a 100% percent of success is reached.

Likewise, the response time (Fig. 10) between the NAO robot indicating the object and the child selecting it was measured. At the beginning, the child took a time of up to 7.5 seconds as he hesitated between several figures before selecting his answer, according to his understanding of geometrical figures and colors increased this time was reduced to a minimum average of 1.8 seconds.

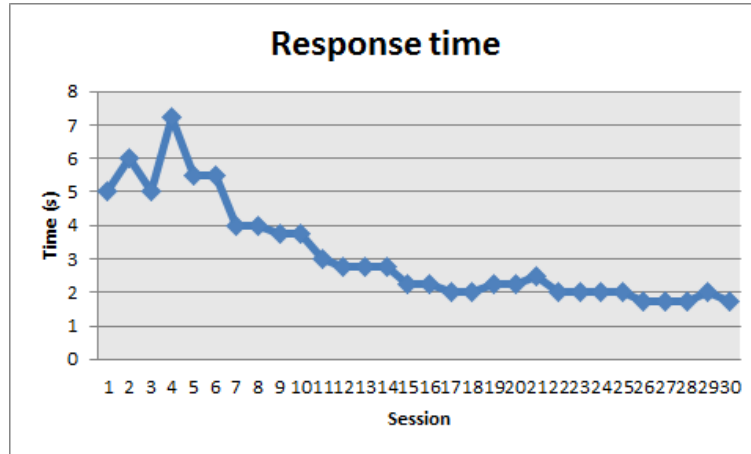


Fig. 10. Response time over sessions.

4 Conclusions

The teaching of concepts to children can be a late process and requires repetitiveness that sometimes can not be provided by a single educator. This paper proposes the use of an NAO robot as a platform for teaching and reinforcing geometric figures and colors to children, being one of the main advantages that the robot can repeat the teaching process as many times as necessary without becoming tired or bored.

In a period of 30 sessions the children demonstrate to have correctly learned the geometrical figures pentagon, square, rectangle triangle and circle, as well the colors red, blue, green, pink and yellow correctly. Being that the game-like teaching seemed interesting to them and they did not show boredom or difficulty to interact with the NAO robot.

Future improvements will focus on expanding the range of colors and geometrical shapes, in addition to introducing the children to concepts such as numbers and letters.

Acknowledgments. H. Sossa would like to acknowledge the support provided by CIC-IPN in carrying out this research. This work was economically supported by SIP-IPN (grant number 20190007). U. Escalona acknowledge CONACYT for the scholarship granted towards pursuing his PhD studies.

References

1. Abuzaraida, M., Jebriel, S.: The detection of the suitable reduction value of douglas peucker algorithm in online handwritten recognition system. IEEE International Conference on Service Operations and Logistics and Informatic SOLI (2015)

2. Beynon, M.: Making new construals of seymour papert's legacy. *Advances in Intelligent Systems and Computing* (2017)
3. Cheng, Y.W., Sun, P.C., Chen, N.S.: The essential applications of educational robot: Requirement analysis from the perspectives of experts, researchers and instructors. *Computers & Education* 126, 399–416 (nov 2018)
4. Conde, B., Fuentes, S., Caron, M., Xiao, D., Collman, R., Howell, K.S.: Development of a robotic and computer vision method to assess foam quality in sparkling wines. *Food Control* Volume 71 (2017)
5. Cunha, J., Ferreira, R., Lau, N.: Computer vision and robotic manipulation for automated feeding of cork drillers. *Materials and Design* Volume 82 (2015)
6. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The canadian cartographer* (1973)
7. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Scopus* (2011)
8. Elkin, M., Sullivan, A., Bers, M.: Programmin with the kibo robotics kit in preschool classrooms. *Computers in the schools*. vol. 33 (2016)
9. Feldman, D., Sung, C., Rus, D.: The single pixel gps: Learning big data signals from tiny coresets. *Massachusetts Institute of Technology* (2012)
10. Hershberger, J., Snoeyink, J.: Speeding up the douglas peucker line simplification algorithm. *Symp on Data Handling* 5th (1992)
11. Keren, G., Fridin, M.: Kindergarten social assistive robot for children's geometric thinking and metacognitive development in preschool education: A pilot study. *Computers in human behavior* (2014)
12. Lopez, P., Sosa, H.: Aprendizaje con y de robotica, algunas experiencias. *Revista Educacion Colombia* (2013)
13. Pinto, M., Barrera, N., Perez, W.: Use of robotics education as a tool in the process of teaching. *Universidad Pedagogica y tecnologica de colombia* (2010)
14. Robles-Bykbaev, V., Andrade-Prieto, E., Solorzano-Guerrero, P., Robles-Bykbaev, Y., Pesantez-Aviles, F., Parra-Astudillo, A.: An educational support tool based on robotic assistants, mobile apps, and expert systems for children with down syndrome. In: *2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*. IEEE (aug 2018)
15. Sandhofer, C., Smith, L.: Why children learn color and size words so differently: Evidence from adult's learning of artificial terms. *Journal of Experimental Psychology* (2001)
16. Senicic, M., Matijevic, M., Nikitovic, M.: Teaching the methods of object detection by robot vision. In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE (may 2018)
17. Smith, S.: *Space and Shape early childhood Mathematics*. SEP (2000)
18. Soendoro, D., Supriana, I.: Traffic sign recognition with color based method, shape arc estimation and svm. *International conference on Electrical Engineering and informatics* (2011)
19. Vigo, L.D., Peyton, R., Bussi, U., Oliva, D.E.: Computer-aided design and simulation of a robotic manipulator for educational purposes. In: *2018 Argentine Conference on Automatic Control (AADECA)*. IEEE (nov 2018)
20. Zhao, Y., Gong, L., Huang, Y., Liu, C.: A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture* (2016)

Transmission and Reception of Text and Images Using Visible Light

Sergio Sandoval-Reyes

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, Mexico
sersand@cic.ipn.mx

Abstract. Communication by light or VLC by its acronym in English (Visible Light Communication), uses visible light from light emitting diodes (LEDs) to transmit information. Using a computing device and some hardware, the transmission of information is performed driving and modulating the light emitted by the LEDs. In the receiver side, the information carried by the modulated light is demodulated through a photo-detector, which is usually connected to a similar computing device for the final recovering of the information. In this article we describe an application based on VLC using OOK (On-Off Keying) modulation, to transmit and receive a line of text and an image from a RaspBerry Pi computer (using Python as the programming language), and several modules (LEDs and a sensor light) from LittleBits.

Keywords: Text and Image Tx-Rx, VLC, OOK, Raspberry Pi, Python.

1 Introduction

Visible Light Communication (i.e. VLC) [1, 2], can be used to transmit audio, voice and data. It requires line of sight between light from emitter diodes (LEDs) and light detectors at the transmitter and receiver ends respectively (Fig. 1). It works in the 380 nm to 780 nm optical band which is visible light and hence the name VLC [3, 4, 5].

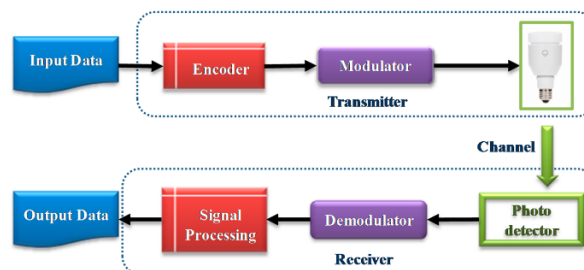


Fig. 1. A basic VLC link structure.

To convey information this one has to be encoded, and then the light has to be modulated and demodulated at the transmitter and receiver sides. There are several methods to do this, some are briefly discussed in the following. Then, the received information

has to be decoded and processed to recover it fully. The success of this recovery depends of several factors, among them: 1) The number, shape, and wavelength of the LEDs employed; 2) The number and type of light detectors (photo-resistor, photo-diode, reverse-biased LED, etc.) used; 3) The encoding method (RZ, NRZ, NRZP, etc.); 4) The modulation scheme (OOK, WPM, VWPM, PPM, OFDM, etc.), and 5) The synchronization and distance between the LEDs and the light detector. Remember that VLC is a technology that among other things, it requires line-of-sight between emitter and receiver [6].

This paper describes an application based on VLC using OOK (On-Off Keying) modulation, to transmit as-a-proof of concept text and color images using a RaspBerry Pi 3 computer as the data source and sink (to simplify the synchronization problem between emitter-receiver), Python as the programming language, and several modules (LEDs and a sensor light) from LittleBits, to easy the hardware implementation.

The remainder of this paper is organized as follows: Section 2 presents a summary of works related to the transmission of text and images using VLC. Section 3 describes the design of our VLC application. Section 4 describe the experiments realized, the results obtained, and their analysis. Finally, our conclusions and future work are presented in Section 5.

2 Related Work

Several research works on VLC technologies to transmit text and images have been proposed. The most important are described in the following.

2.1 How Based VLC Text and Image Systems Work

Typical based VLC image systems are implemented using an intensity modulation and direct detection (IM/DD) scheme with a line-of-sight (LOS) configuration [7]. In the transmitter, IM is implemented through the modulation of the transmitted signal into the instantaneous optical power of the LED by controlling the radiant intensity with the forward current through the LED (High modulation frequencies are used to avoid flicker). In the receiver, the transmitted signal is recovered using direct detection (DD). In this simple method, a photodiode (or array) is used to convert the incident optical signal power into a proportional current. Figure 2 shows a general flow chart for VLC text and image transmission [8].

2.2 VLC Text and Image Transmitter

A typical based VLC text and image transmitter contains a text and image generator (a PC with Matlab to convert the text and the image into bits), an interface (usually a USB cable) to send the bits to a microcontroller (for coding and modulation), and taken through one of its output ports to the LED driver and the LED optics, see Figure 3 [9]. The modulated signals are used to switch on-an-off the LEDs at desired frequencies using LED drivers. These drivers rely on transconductance amplifiers to convert voltage signals into corresponding current signals to excite the LEDs array for communication purposes.

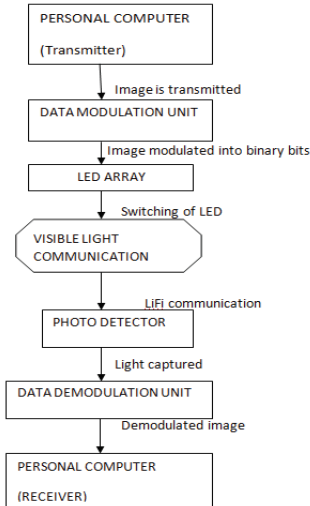


Fig. 2. A flow chart for VLC text and image transmission.

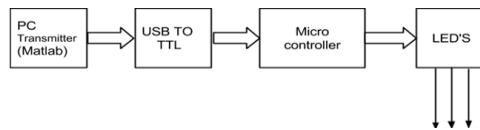


Fig. 3. VLC text and image transmission.

2.2.1 VLC Modulation

Although there are different modulation schemes for VLC, mainly, on-off keying (OOK), variable pulse-position modulation (VPPM), color shift keying (CSK) and orthogonal frequency division multiplexing (OFDM) [10], OOK is the most popular. OOK is the most commonly used IM/DD modulation scheme in VLC due to its simple implementation. In this method, basically the LED intensity is changed between two distinguishable levels corresponding to the data bits (1 or 0), see Figure 4. A modified OOK, called Variable OOK (VOOK) can provide dimming. It is achieved by changing the data duty cycle through pulse-width modulation (PWM), with only 1 bit of information carried per symbol period.

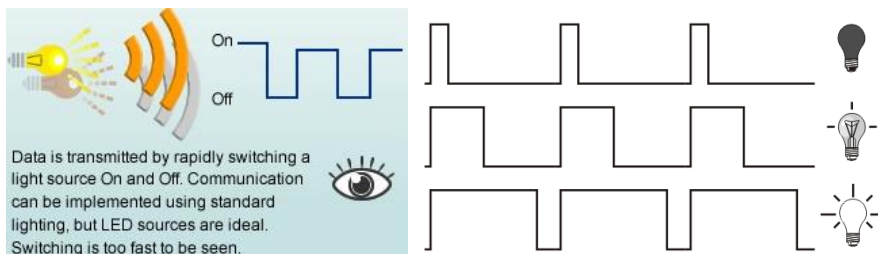


Fig. 4. On-Off Keying with PWM.

2.3 VLC Text and Image Receiver

A typical optical text and image receiver consists of a photo detector followed by an amplifier. The photo detector can be a photo transistor, a reverse-biased LED, or a Light Detect Resistor (LDR). The light captured by the photo transistor which acts as a sensor, passes the output to the comparator which compares the binary input, and similarly the original image is recovered using Matlab software in the PC, see Figure 5 [8, 9].

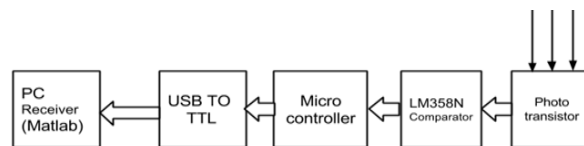


Fig. 5. VLC text and image reception.

3 Image Transmission and Recovering using VLC

In the following, we develop a VLC application to transmit and receive in real-time first a line of text, and then an image using two LEDs in the transmitter, and a photo detector in the receiver.

In order to do that, we will use in the transmitter as a data source, a Raspberry Pi 3 (RBPi) computer, and five LittleBits bit-modules [12]: power, button, proto, split, and two bright LED bits, see Figure 7. The RBPi using a software written in Python will read an image byte by byte, and will send each byte using the SPI MOSI (Master Output Slave Input) output port (Pin 19), toward the proto module, OOK signals to drive the two LEDs.



Fig. 6. Receiver components: Light sensor and proto bits.

While in the receiver, we will use three LittleBits components: power, light sensor, and another proto bit, see Figure 6. The light sensor captures the light emitted by the two LEDs and converts it into a digital signal, which is fed to the proto module. The proto module in turn outputs this signal and with a wire connector, feeds this signal toward the MISO (Master Input Slave Output) input port (Pin 21) of the RBPi.

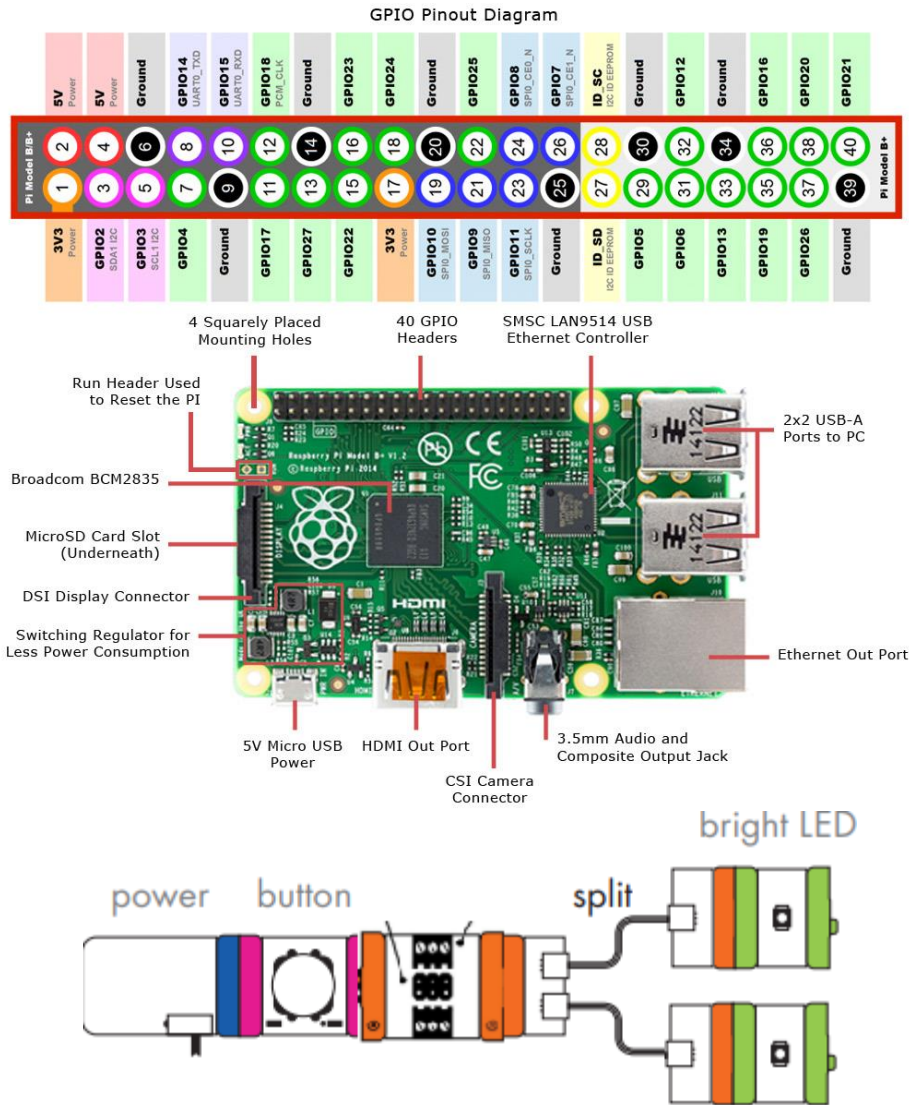


Fig. 7. Raspberry Pi 3, and five bits: power, button, proto, split and two LEDS.

3.1 Transmitting a Line of Text and an Image via VLC

The program *hola_lifi.py* written in python transmits a line of text, see Figure 8a.

```
# hola_lifi.py
# Python code to transmit and recover the line of text "hola Li-Fi" using SPI and VLC
import spidev
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz = 4000000
buffer = bytearray()
```

```
tx_data = [ord("H"), ord("o"), ord("l"), ord("a"), ord(" "), ord("L"), ord("i"), ord("-"), ord("F"),
ord("i")]
while True:
    # Transmit the text "Hola-Li-Fi", through the Raspberry out pin 19 SPI-MOSI,
    # and receive it in the array rx_data, through the Raspberry input pin 21 SPI-MISO
    rx_data = spi.xfer(tx_data)

    # Receive every byte in rx_data and store them in a buffer to display them in the screen
    for byte in rx_data:
        buffer.append(chr(byte))
    print buffer
```

Fig. 8a. Python code to transmit and recover a line of text via VLC.

To transmit an image via VLC, the RBPi with a script written in Python, opens a picture file ("lenna.jpeg") and it reads it into a bytearray "b". Then with a "for" loop reads byte by byte of the picture and it send them to the SPI MOSI output port 19, using the spi.xfer () directive. This MOSI output is fed using two wire connectors (signal and ground), into the input of the "proto" module (lower middle connector in figure 7). The proto module outputs and split the OOK signals to drive the two LEDs. The Python code to execute the above mentioned is shown in Figure 8b.

```
# lena_spy.py: Python code to read, transmit, and recover a picture via Visible Light
import spidev
from PIL import Image, ImageFilter
import io
from array import array
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz = 4000000
b = bytearray()
buffer = bytearray()
try:
    # Load an image from Raspberry Pi
    with open ("lenna.jpeg", "rb") as img:
        f = img.read()
        b = bytearray(f)
    # Sending with SPI and converting the image, byte by byte into visible light
    for byte in b:
        rx = spi.xfer([byte])
        rx_data = rx[0]
        buffer.append(rx_data)
    # Recovering the bytes array and back into an image
    img_rec = Image.open(io.BytesIO(buffer))
    img_rec.save('lena_img_recovered.png')
    img_rec.show()
except:
    print "Unable to recover image"
```

Fig. 8b. Python code to read, transmit and recover a picture via VLC.

In this code, it is necessary to import the following libraries: "SPI", "PIL" (Python Image Library), and "Array". The use of the LittleBits modules simplified very much the hardware implementation. The power module was fed with a 9-V battery, and this was necessary because the outputs of the RBPi are low-voltage (3.3 volts) and low-current (Individual pins must not pull more than 16 mA and the entire GPIO must not source more than 50 mA), which are no good enough to drive two bright LEDs [13]. These LEDs are simple yellow LEDs with a wavelength of 550-to-600 nm, luminous flux of

Research in Computing Science 148(10), 2019 340 ISSN 1870-4069

4-to-5 lm, and consume around 16-to-20 mA each, with an aperture angle of about 120 degrees, see figure 9.



Fig. 9. LittleBits bright LED.

3.2 Receiving Text and an Image via VLC

As it was mentioned, the picture was sent via VLC as LED light. This light is received through a light sensor module which then sent it back through out another proto module, to the MISO port 21 of the RBPi. This light sensor not only receives the OOK light signal but also has a transimpedance amplifier for high speed operation.

The light sensor has 2 modes, Figure 7. In LIGHT mode, as the light shining on the sensor gets brighter, more signal passes through it. In DARK mode, the signal increases as it gets darker. Furthermore, the light sensor has a sensitivity dial or slide dimmer to adjust how much light it takes to change the signal, and has a spectral sensitivity range from 500-to-600 nm similar to the LEDs wavelength, see Figure 10.

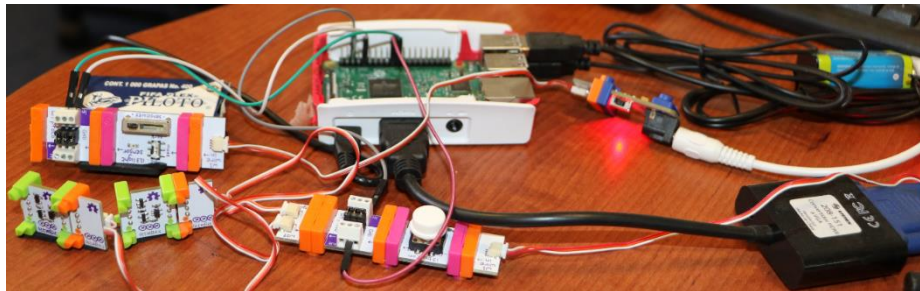


Fig. 10. Receiving text and a picture via VLC.

The recovered text and picture from the SPI MISO input is stored into a buffer, saved and displayed, as can be seen from the Python code lines of figure 8a and 8b.

4 Experiments and Results

For the experiments, we use as was mentioned a RaspBerry Pi 3 computer with several LittleBits components. The whole setup is shown in Figure 11.



Fig. 11. Setup for the transmission and reception of text and an image using VLC.

```
pi@raspberrypi ~  
File Edit Tabs Help  
ola Li-Fi  
ola Li-Fi  
Hola Li-Fi  
ola Li-Fi  
Hola Li-Fi  
Hola Li-Fi  
o  
ai-Fi  
[Ctrl]C  
Traceback (most recent call last):  
  File "hola_lifi.py", line 24, in <module>  
    print buffer  
KeyboardInterrupt  
pi@raspberrypi:~$ sudo python lena_spi.py  
255 216  
pi@raspberrypi:~$
```

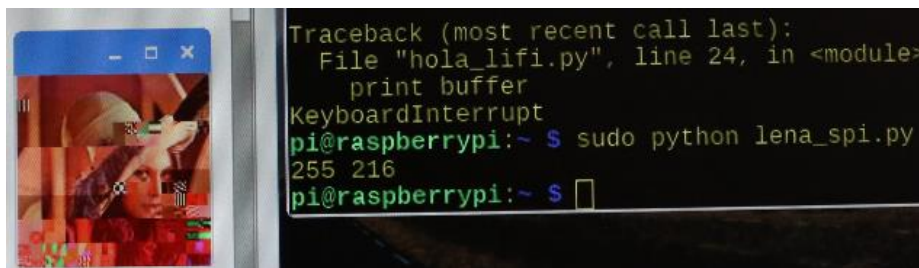


Fig. 12. Recovering of text and picture via VLC.

Figure 11 shows the RaspBerry Pi 3 computer connected through out bus connections to a HP display, keyboard and mouse. Figure 12 shows the recovered text and picture after the execution of the line commands “*sudo python hola_lifi.py*” and “*sudo python lena_spi.py*”, see Figure 12.

Figure 12 also shows that the recovered text and picture was no perfect. That was due to the presence of noise, mainly: Fluorescent light from ceiling lamps, misalignment and distance between LEDs and light sensor, and low sensitivity to light from the sensor.

4.1 Discussion of Results

After several adjustments to the setup and a few intents, the text and the picture was finally well recovered. The whole transmission and recovering of a 225 x 225 pixel color image weighting 8 Kbytes, took less than a second, see Figure 13. Naturally, the noise cannot completely eliminated and increases when the misalignment and the distance between the LEDs and the light sensor is larger. In addition, the brightness of the LEDs influence the performance. That was the main reason for using two LEDs in parallel to increase the amount of light sent to the light sensor.



Fig. 13. Recovered picture via VLC after some adjustments.

4.2 Metrics of the Recovered Text and Picture via VLC

Because the transmission (LEDs) and reception (light sensor) link is made using RaspBerry SPI, two questions arise: 1) What percentage of the image was recovered correctly? and 2) How many times the image was well recovered, versus the number of times the image could not be recovered?

1) Percentage of the image that was recovered correctly: As it was showed in figure 12, initially without any kind of adjustments, the recovery of the image was around 50 %. Notice from this figure the pixelation of the image in certain areas. In contrast with the adjustments mentioned before, the image was 100 % recovered as it is shown in figure 13. Similar results were obtained for the line of text.

2) Number of times the image was well recovered: Even though the adjustments performed to the transmitter-receiver setup, the link was shaky, requiring many tries to obtain a 100 % image recovery. The number of times the image was well recovered, was 3-out-of-10. In contrast, the recovery of the line of text required less tries.

4.3 Contributions

This work differs with respect to similar works as in [8], [9], [11] and [14], in the following. In [8] only a design model is proposed without any proof of implementation. In [9], a hardware setup is showed, but again, there is not any proof of sending and receiving any picture. It is also required to use two microcontrollers (one for the transmission and one for the reception). The main problem of using microcontrollers is that not all of them have enough RAM memory to store a medium to large image sent by the PC. Furthermore, the Matlab software have to reside at both PCs to connect with the microcontrollers and to process the picture downloading and uploading. In [11] a Windows PC with Matlab and an Arduino Uno microcontroller is used for the serial transmission and reception of two 24 x 25 color pictures, but without using visible light communication. In [14], the image transmission requires infrared light.

5 Conclusions

A line of text and an image transmission and reception application using VLC was developed using a Raspberry Pi 3 computer, two bright LEDs and a light sensor from LittleBits, OOK modulation, and the Python Image Library. Both the line of text and the picture in color, were recovered acceptably well although with a small presence of noise. It should be noted that this noise is due to environment light, and the distance between the LEDs and the light sensor.

Also, the performance of the application depends on the brightness of the LEDs and the sensitivity of the light sensor with respect to the light received from the LEDs and the surrounding light (in Figure 7 it can be seen that the light sensor has a control for graduating the sensitivity in the presence of high or low light). Additionally, the alignment between the LEDs and the light sensor influences the reception, and consequently the quality of the image reproduction.

5.1 Future Work

This application could be improved by: 1) Increasing the number and/or power of the LEDs to increase the reception distance; 2) Using better modulation schemes; and 3) Implement and include a continuous synchronization mechanism and error correcting codes, to improve the LED-to-light sensor VLC transmission-reception.

Acknowledgments. This work was supported by Instituto Politecnico Nacional Project SIP-IPN 20196133.

References

1. TED Ideas worth spreading. Harald Haas: Wireless data from every light bulb. http://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb (2011)
2. Dobroslav, T., Videv, S.; Haas, H.: Light fidelity (Li-Fi): towards all-optical networking. In: Proc. SPIE (Broadband Access Communication Technologies VIII) 9007 (2) doi:10.1117/12.2044649 (2013)
3. Sherman, J.: How LED Light Bulbs could replace Wi-Fi. Digital Trends. <http://www.digitaltrends.com/mobile/light-bulb-li-fi-wireless-internet/> (2013)

4. Haas, H.: High-speed wireless networking using visible light. SPIE Newsroom. doi:10.1117/2.1201304.004773, <http://www.spie.org/newsroom/4773-high--speed-wireless-networking-using-visible-light> (2013)
5. Vincent, J.: Li-Fi revolution: internet connections using light bulbs are 250 times faster than broadband. <http://www.independent.co.uk/news/science/li-fi-revolution-internet-connections-using-light-bulbs-are-250-times-faster-than-broadband-8909320.html> (2013)
6. Wikipedia: Location awareness. https://en.wikipedia.org/wiki/Location_awareness (2016)
7. Jovicic, A., Li, J., Richardson, T.: Visible light communication: opportunities, challenges and the path to market. DOI: 10.1109/MCOM.2013.6685754 <http://ieeexplore.ieee.org/document/6685754/> (2013)
8. Mahendran, R.: Integrated Lifi (Light Fidelity) For Smart Communication through Illumination. In: International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp 53–56. DOI: 10.1109/ICACCCT.2016.7831599 <https://ieeexplore.ieee.org/document/7831599/?reload=true> (2016)
9. Vyom, S. et al.: 2D Image Transmission using Light Fidelity Technology, Communication. International Journal of Innovation & Advancement in Computing Science, IJIACS, 4(4), 121–126, <https://www.academicscience.co.in/admin/resources/project/paper/f201504241429887963.pdf> (2015)
10. Kwonhyung, L.; Hyuncheol, P.: Modulations for Visible Light Communications with Dimming Control. IEEE Photonics Technology Letters 23(16) DOI: 10.1109/LPT.2011.2157676 <http://ieeexplore.ieee.org/document/5773477/> (2011)
11. Cubas Perfecto, G., Santiago Godoy, R., Anzueto Rios, A.: Transmisión de imágenes de Matlab a Arduino vía Puerto Serial. Boletín UPIITA-IPN, 644 CyT, No. 52 <http://www.boletin.upiita.ipn.mx/index.php/ciencia/644-cyt-numero-52/1234-transmision-de-imagenes-de-matlab-a-arduino-via-puerto-serial> (2016)
12. LittleBits: <https://www.littlebits.cc/> (2017)
13. Raspberry Pi: Raspberry Pi input and output pin voltage and current capability. Mosaic Documentation Web, <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications> (2018)
14. Mohit, V., Amritansh, K., Mishra, U., Singh, C., Deeksha, C., Shashank, K.: Image Transmission Using Li-Fi. In: IEEE-ICICCT <https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Image%20Transmission%20Using%20Li-Fi> (2018)

Voice Synthesis System Based on Recursive Functions Designed by Graphs

Rodolfo Romero Herrera¹, José Elias Romero Martínez²

¹ Instituto Politécnico Nacional, Escuela Superior de Cómputo,
Departamento de Ciencias e Ingeniería de Computación,
Ciudad de México, Mexico

² Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica
Ciudad de México, Mexico
rromero@ipn.mx, saile14398@gmail.com

Abstract. A recursive function based on the use of graphs is presented. The recursive equation is based on mathematical morphology; A written text divided into syllables is related to phonemes of spoken Spanish in Mexico. The algorithm is implemented in a PIC and an integrated circuit that stores the sounds in memory locations. Each space in memory becomes a code that is related to a phoneme. Thus, the recursive function receives as input a vector that is the structure of a word; and outputs the corresponding memory locations. Obtaining intelligibility of more than 84%. The development of the recursive function in graphs is proposed, which allows an easy description of the sequence because it is handled as a black box with input and outputs; and the equation that governs the system can even be a transfer function.

Keywords: Synthesis of voice, graph, recursion, mathematical linguistics.

1 Introduction

Recursive functions are observed and treated in different ways. In the case of artificial intelligence, the recognition of patterns is a case treated by many [1]; and of course, hardware designed with the appearance of FPGAs have a special meaning in the design; although it is complicated to implement, especially when it is not accessible to most programmers because of the specialization required [2]. A useful option is to visualize the systems as black boxes, considering their inputs and outputs, as well as their transfer function; however, some form of describing them is required. What generates matrices that allow identifying states and exits of the system that consequently takes us to recursive functions that can even depend on probabilities [3]. This is where the description of graph-based systems simplifies the design and application of even nature-based systems such as Genetic Algorithms [4]; or problems of analysis of public transport networks or interconnected social networks, etc. [5 to 7].

This establishes the design of systems based on graphs that can be directed or not, and that allows establishing a correspondence between even the combinational design [8], sequential or systems based on probabilities and the algebra of graph theory [9] [10]. In this way, in the present work, a voice reproduction system based on phonemes

is designed using the graph technique for the recursive function; and mathematical linguistics to describe the grammar of function [11].

2 Voice Processing

A digital signal processing project is presented that justifies the results of the methodology proposed in this work. The project consists of a phonological system of the Spanish language spoken in Mexico.

For the development of the project, the naturalness and intelligibility of the resulting voice and mainly the complexity of the processing was taken into account, because the use of a graph-based technology is tested [13]. Once proven its feasibility, it is feasible not only to use phonemes, it is also possible to use di-phonemes or the combination of both.

For the recording of the phoneme, the WAV format was used with a sampling frequency of 11025Hz, mono-stereo, at 16 bits [11]. The following steps were carried out with Matlab:

- Filtering the phoneme,
- Elimination of entrance noise,
- Delimitation of the phoneme,
- Phoneme selection.

The recording of all the simple combinations with a vowel and complex with existing vowel was performed, as well as separate vowels, and consonants located in coda position susceptible to being isolated and recorded, such as “*b, c, d, f, g, l, m, n, r*” and “*s*”. The syllables that have the consonants *p* and *t* in coda position are reviewed and recorded together with the attack (if it exists) and the nucleus [14].

Table 1. Phonemes in coda position.

Simple attack	Complex attack with l	Complex attack with r	Core
<i>b [a - u]</i>	<i>b [a - u]</i>	<i>br [a - u]</i>	<i>A</i>
<i>c [a - u]</i>	Does not apply	Does not apply	<i>E</i>
<i>d [a - u]</i>	Does not apply	<i>dr [a - u]</i>	<i>I</i>
<i>f [a - u]</i>	<i>fl [a - u]</i>	<i>fr [a - u]</i>	<i>O</i>
<i>g [a - u]</i>	<i>gl [a - u]</i>	<i>gr [a - u]</i>	<i>U</i>
<i>j [a - u]</i>	Does not apply	Does not apply	
<i>k [a - u]</i>	<i>kl [a - u]</i>	<i>kr [a - u]</i>	
<i>l [a - u]</i>	Does not apply	Does not apply	
<i>ll [a - u]</i>	Does not apply	Does not apply	
<i>m [a - u]</i>	Does not apply	Does not apply	
<i>n [a - u]</i>	Does not apply	Does not apply	
<i>ñ [a - u]</i>	Does not apply	Does not apply	
<i>P [a - u]</i>	<i>pl [a - u]</i>	<i>pr [a - u]</i>	
<i>r [a - u]</i>	Does not apply	Does not apply	
<i>rr [a - u]</i>	Does not apply	Does not apply	
<i>s [a - u]</i>	Does not apply	Does not apply	
<i>t [a - u]</i>	<i>tl [a - u]</i>	<i>tr [a - u]</i>	

Table 1 shows some phonemes in coda position, given the way in which they are generated, specifically p and t; it was not possible to isolate them in a way; so it is necessary to record the basic combinations of Spanish that involve these two elements in coda position [14].

3 Mathematical Morphology for the Voice Concatenation Function

We used a microcontroller that implements the equation of nature [14]. See equation (1) where S is a black box, e is an element, e * means that you can have as many elements as you want and S * indicates that the box can be called as many times as you want. If S is a speech synthesizer system to produce any word of Spanish, it can call itself as many times as the limit that is marked by a phrase.

$$S \rightarrow e^*S^* \tag{1}$$

Mathematical linguistics was used in the speech synthesizer. If it works by mathematical linguistics, it carries out the following actions:

El orador dice hola,
El orador dice Adios.

These sentences work with a structure of sentences seeking to find a general structure or syntactic rule from particular structures or canonical prayers. An equivalent sentence is:

El orador dice hola y dice Adios

Now fragments of the sentence are replaced by labels according to Table 2; where $o_1r_1o_2$ and $o_1r_1o_3$ in equation 2 are known as canonical sentences [16].

Table 2. Replacement of fragment by label.

Fragment	Label
<i>El orador</i>	o_1
<i>Dice</i>	r_1
<i>Hola</i>	o_2
<i>Y</i>	+
<i>Dice</i>	r_1
<i>Adios</i>	o_3

$$o_1 r_1 o_2 + o_1 r_1 o_3 \tag{2}$$

The algebraic factorization consists of finding the common factors in an expression. Applying the factorization to the canonical sentences of (2) we have to:

$$o_1 r_1 (o_2 + o_3) \tag{3}$$

In (3) the simplification of words is observed, so that memory is saved and it is possible to implement it as a recursive function. When a set of sentences contains an element that is repeated indefinitely, it is convenient to replace the entire set of sentences with a recursive grammar, for example:

$$S \rightarrow ab; aab; aaab; aaa...ab; \tag{4}$$

An associated recursive grammar is:

$$S \rightarrow aS | b, \tag{5}$$

where "|" represents the concatenation. Substituting S for aS or for b can have chains like the following:

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaab. \tag{6}$$

This can be understood as a system or reusable component that is called likewise. Recursion is introduced when it is detected that a string "a" repeats monotonously. For example, in prayer:

$$Abcabcbcabcd, \tag{7}$$

where the string *abc* presents a monotonous behavior that ends in d.

Equation 4 and 6 can be represented with directed graphs. See figure 1.

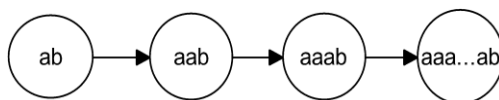


Fig. 1. Graph equation 6.

But as shown in (5) it is feasible to modify the graph in such a way that it can be implemented as a directed graph.

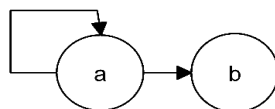


Fig. 2. Graph equation 5.

4 Voice Processing

Once defined the elements and the mathematical foundation of the system, you can establish the graphs that allow the design of the system, which will depend on the degree of abstraction that the designer determines convenient since this can be very specific or very generic. Figure 3 presents the graph for the synthesizer where each node represents a function.

- a) The solution of the irregularities of the Spanish language applied to all the phrases and substitution of graphemes by the symbols used in the representation of phonemes in the speech synthesizer.
- b) Separation in syllables of the sentence according to syllabication rules.
- c) Extraction of phonemes from the phrase.
- d) Concatenation of the samples of the audio files of the phonemes to be used.
- e) Reproduction of the samples of the resulting concatenation.

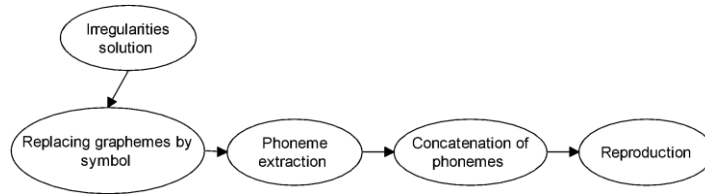


Fig. 3. Graph for components of the synthesizer.

The Conca function is responsible for concatenating the different files stored in a single object, which is responsible for reproducing the input message. Once the arrangement with the names of the audio files has been obtained, Conca opens each of these files and its samples are read.

Samples are stored in an object that contains the samples of all the files (each file contains a phoneme). When you finish writing this object, you start writing the samples of the next file, until you finish with the last one. Figure 4 presents the graph corresponding to Conca.

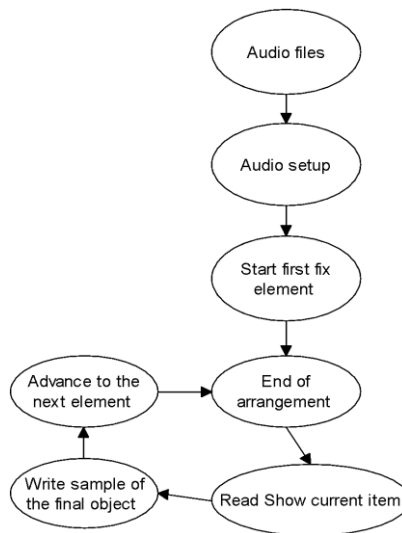


Fig. 4. Degree of Conca.

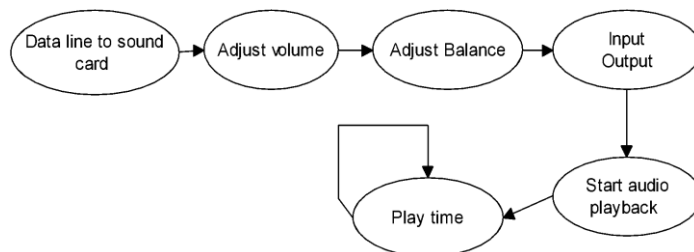


Fig. 5. Degree of the Play function.

Figure 5 shows the graph of the Play function. The created phrase is taken and reproduced. It is also possible to adjust the volume and balance of playback by two parameters in this module.

5 Results

The voice obtained at the output of the synthesizer has an intelligible average quality. Figure 6 (a) shows the shape of the signal from the word lomo recorded in a normal way. Figure 6 (b) presents the concatenation of phonemes. The decrease and increase of the signal between the combination of the phoneme “lo” and “mo” are clearly observed. This reduction creates a lack of fluidity in the resulting voice.

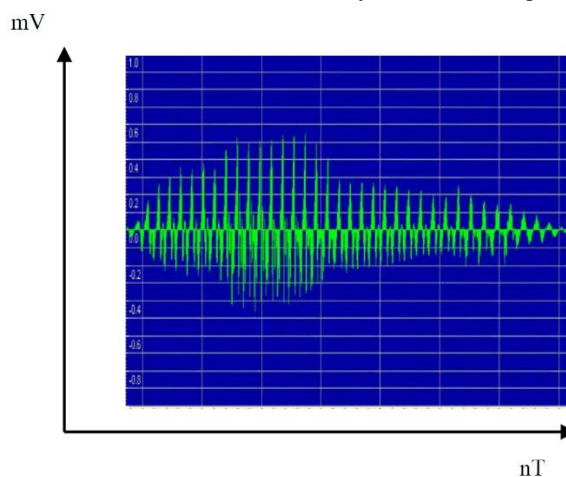


Fig. 6. (a) Word "lomo" recorded.

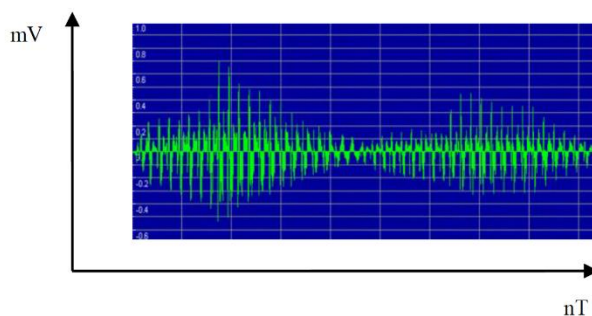


Fig. 6. (b) Word "lomo" synthesized.

The intelligibility test of the speech synthesizer was applied with a group of people to listen to generated words and phrases, without previously knowing what they were [17]. The results of these tests are shown in Tables 3 and 4, respectively. From these tests, it is deduced that only 15.66 did not understand the phrase; 50 male and 50 female individuals were taken to perform the tests.

Table 3. Intelligible phrases.

Words understood correctly:	253
Words that are incorrectly understood:	47
Intelligibility percentage:	84.34 %

For the second test, phrases were chosen that did not have something in common, but that did have coherence by themselves. Only 0.5% did not understand the phrase.

Table 4. Intelligibility in sentences without something in common.

Phrases correctly understood:	224
Phrases included incorrectly:	1
Intelligibility percentage:	99.55%

Figure 7 shows the physical system of voice reproduction, where the components are observed. The functions seen as black boxes were implemented in hardware through an AX PIC. The circuit only uses the PIC together with an audio storage memory where the different phonemes to be reproduced are placed [18].



Fig. 7. Voice synthesizer system.

6 Conclusions

A proposal for modeling in graphs was presented. The proposed diagrams are based on the object-oriented design, which will allow us to obtain efficient graphics modeling. In the same way, we present how mathematical linguistics allows us to represent systems through the equation of nature; checking how a reusable system can be called or multiplied to develop complex tasks, as a recursive function does, proposing the construction of hardware systems supported by concatenation operations. The system applied to hardware development allows us to simplify resources and take advantage of the potential of devices such as microprocessors, etc. In such a way that a system in hardware can call itself so many times until it reaches the limit. Seen in this way, generic systems represented by a graph can be produced. Where an object can be a component or a reusable system and this, in turn, a recursive function in hardware.

Although no tests with di-phonemes were performed, the intelligibility level exceeded 99%. In future work, in order to avoid the robotic voice, it is considered to make an integration of phonemes and phonemes; once it has been demonstrated that the design using graphs gives advantages; Although no tests with di-phonemes were performed, the intelligibility level exceeded 99%. In future work, in order to avoid the robotic voice, it is considered to make an integration of phonemes and phonemes; once it has been demonstrated that the design using graphs gives advantages; since, although here it was not considered conditional probabilities or some other factors it is evident that when applying it results in greater benefits.

Acknowledgment. Acknowledgments to the IPN for the support received for the development of this Project.

References

1. Puerto Cuadros, E.G., Aguilar Castro, J.L.: Un algoritmo recursivo de reconocimiento de patrones. *Revista Técnica* 40(2), p. 95 (2017)
2. Pimentel, B.F.: *Synthesis of FPGA-Based Accelerators Implementing Recursive Algorithms*. ProQuest Dissertations Publishing (2009)
3. Medel Juárez, J.J., Zagaceta Álvarez, M.T.: Estimación-identificación como filtro digital integrado: descripción e implementación recursiva. *Revista Mexicana de Física* 56(1), 1–8 (2010)
4. Quintero, J.A.G., Toledo, E.S., Diaz, M.A.: Reconfiguración de sistemas de distribución mediante algoritmos genéticos basados en la teoría de Grafos. *Ingeniería Energética* 37(2), p. 115 (2016)
5. Cardozo, O.D., Góme, E.L., Parras, M.A.: Teoría de grafos y sistemas de información geográfica aplicados al transporte público de pasajeros en resistencia (Argentina). *Transporte y Territorio*, No. 1, pp. 89–111 (2009)
6. Mena Díaz, N.: *Redes sociales y Gestión de la Información: un enfoque desde la teoría de grafos* (2012)
7. Escalona, Y.E. *et al*: Sistema para la solución de problemas de cómputo basado en el Método de los Grafos Dicromáticos. *GECONTEC: Revista Internacional de Gestión del Conocimiento y la Tecnología* 1(1), 37–46 (2014)
8. Avalos Gaytán, V., Rivera Ramírez, M., Schaeffer, E.: *Agrupamiento local en grafos dirigidos* (2009)
9. Haemers, W.H., Kharaghani, H., Meulenberg, M.A.: Divisible design graphs. *Journal of Combinatorial Theory, Series A*, 118(3), 978–992 (2011)
10. Booch, G.: *The Unified Modeling Language User Guide*. Addison-Wesley, USA (2000)
11. Matworks: Matlab: <https://la.mathworks.com/products/matlab.html>, USA (2019)
12. Ortega, A. et al: Graph Signal Processing: Overview, Challenges, and Applications. *Proceedings of the IEEE* 106(5), 808–828 (2018)
13. Voiers, W.D.: Diagnostic Evaluation of Speech Intelligibility. In: Hawley, M.E. (ed.): *Speech Intelligibility and Speaker Recognition*, Dowden, Hutchinson, and Ross. Stroudsburg, PA. (1977)
14. Valero, L.A.: *Curso audiovisual interactivo de codificación de voz*. Departamento de electrónica y Tecnología de computadoras, Universidad de Granada, <http://ceres.ugr.es/~alumnos/luis/codif.htm>, España (2005)
15. Galindo Soria, F.: *Una ecuación de la Naturaleza*. Escuela Superior Cómputo, Instituto Politécnico Nacional (1998)

16. Marcus, S., Nicolau, E., Stati, S.: *Introducción en la lingüística matemática*. Editorial Teide/Barcelona (1966)
17. Voiers, W.D.: *Diagnostic Evaluation of Speech Intelligibility*. In: Hawley, M.E. (ed.): *Speech Intelligibility and Speaker Recognition*, Dowden, Hutchinson, and Ross. Stroudsburg, PA. (1977)
18. Piacaxe, <http://www.picaxe.com/> (2019)

When Twitter Becomes a Data Source for Geospatial Analysis

Magdalena Saldana-Perez¹, Camille Cavalière², Miguel Torres-Ruiz¹,
Marco Moreno-Ibarra¹

¹ Instituto Politécnico Nacional, Centro de Investigación en Computación,
Laboratorio de Procesamiento Inteligente de Información Geoespacial, Mexico
² University of Grenoble Alpes, Laboratoire d'Informatique de Grenoble, France

{amagdasaldana, mtorres, marcomoreno}@cic.ipn.mx,
Camille.Cavaliere@imag.fr

Abstract. Social media has become a useful data source for processes and researches interested in improving people's life. Publications done by social media users provide details about the people perceptions of their environment, and updated observations about what happens in real world. In this approach the relevance of Twitter as a data source for scientific purposes is analyzed, as well as its use in geospatial researches. Tweets have two main characteristics, a text where user describes its ideas, and metadata, where features such as the coordinates of the place where the tweet was posted are stored. Different computing procedures are applied over tweets in order to make them useful for different tasks; commonly, text mining, classification, and regression algorithms are used to process tweets. The coordinates of tweets make possible to link events described in tweets with the geographical area where they occur. The analysis of tweets and coordinates provides updated data, useful in natural disasters control, decision taking processes and urban studies. The present approach studies what motivates users to tweet, and analyzing the messages produced by Twitter users, classifies the tweets into three groups according to the information expressed in their text: *self-centered*, *social information* and *collective information*. Additionally, some methods to extract information from tweets are studied, common problems presented when working with tweets, and some researches that use them in a geospatial domain are presented.

Keywords: twitter, social media, data treatment, classification, text mining, geospatial analysis, data problems.

1 Introduction

Twitter is commonly used as data source at scientific and commercial data analysis; the microblogging features are analyzed by researches interested in emergency information diffusion, on the dissemination of news, and on the way in which authorities and civil societies help citizens with problems or crisis situations [45].

The microblogs posted on Twitter are called tweets, they have two main elements: metadata, and a text limited to 280 characters; their metadata is composed of a timestamp (the date and time of the tweet's publication), the coordinates of the place

where the tweet was posted, and the id of the user who wrote the tweet [71]. In Twitter, users are able to see some others tweets, acting as followers. Users can share their tweets with all the media members (public profile), or with specific users (private profile) [3].

Considered as a new data source, Twitter provides people's points of view from many different topics such as health, environment, politics, economy, sports, and natural disasters, among others. The social media lets researchers know the people's feelings and ideas about factors that disturb their daily activities; governments and institutions interested in improving people's lives and in solving specific problems can use its updated information [57]. Twitter facilitates the gathering of crowd-sourced information [46]. Twitter provides information in less time than the methods used to collect data before the social media explosion. In the past, data related to people's feelings and opinions about their environment were collected applying surveys and census [34].

The data processing of the tweets text and metadata components becomes a challenge, sometimes users do not allow the social media to know their location, tweets do not have coordinates. Sometimes people just tweet comments about a topic, this represents a problem for scientific researches since most of the times personal comments do not provide relevant information [42]. For example, when terrorism acts occur special hashtags are created in order to group all the tweets related to the act; many tweets use hashtags, some of them provide information about the event, and some others describe people's feelings (these last generate noisy information). Noisy tweets affect the quality of the tweets sample when it is processed, personal comments must be filtered in order to assure that the collected tweets contain valuable information. When tweets regarding to a specific topic are tweeted far from the study area, is important to consider the variations of space and time between them. In developed countries, citizens volunteering and the information from social media or collective systems without an official agency affiliation are undervalued (Whittaker et al., 2015). The use of extra data sources when working with Twitter helps validate the tweets data and ameliorate the researches accuracy [67].

According to Haworth (2016), the citizens observing, collecting, sharing and analyzing data have led the development of many scientific researches that would not have been possible otherwise. Similarly to the collaborative information and social media data, the quality of citizen's science has been questioned, but it has been shown that applying the adequate preprocessing procedures, the citizens' data can meet the same quality as data collected from official sources [10]. People's participation has lead the citizen science, which refers to engage public citizens in scientific research projects [7].

2 Objective

The present approach analyses the use of Twitter in scientific researches related to the geospatial domain. The objective of the approach is show the importance and utility of data obtained from Twitter when are used in the geographic domain, and propose a classification of tweets based on the information they provide for different data analysis. The approach presents the relevance of data extracted from tweets, the

methods used to manage them, and some problems presented when working with the most famous microblog service. In addition, research works that merge tweets with geographic information systems (GIS) are described.

The paper is organized as follows, in section two are presented the material and methods of the investigation, in section three the theory of the approach and the motivation of people to tweet are analyzed, in section four results and discussions are presented; finally, the conclusions of the approach are given.

3 Material and Methods

Volunteered geographic information (VGI) refers to the wide creation of geographic information by citizens using web platforms, free mapping tools, mobile applications and social media. VGI has changed the creation and propagation of information generated by people [34].

In disaster management context VGI can also be labelled as digital volunteering (McLennan et al., 2015), or digital humanities [13]. Digital volunteering has become a source of asserted information used to complement geographic information from governmental agencies and private organizations [37].

Many researching works that use Twitter as a data source meet in one point: the data provided by tweets is an updated vision about people's perceptions of their environment. Landwehr et al. (2016) see in Twitter the opportunity to get information out of public, which increases the rate of dissemination. According to Haworth and Bruce (2015), VGI academic researches and emergency management studies have focused on citizens' science. For example, the 2010 Haiti earthquake where volunteers from all over the world worked together to map the affected area [51]; the volunteered mapping of the Nepal earthquake crisis [34] and the crowdmaps responding to the cyclone and floods in Queensland [49]. Another valuable example of emergency management using citizen's data is the OSM tasking manager [54], a mapping tool designed for the Humanitarian OSM Team to give solutions when critical situations occur, such as the Hurricane Maria (20/09/2017) and the Mexican Earthquake (19/09/2017).

Some studies emphasize the benefits of VGI and social media data meanwhile some others argued that web platforms marginalize people without internet and technology access. Benefits of VGI are the timely information exchange and the connectedness (Taylor et al., 2012), the provisional information for disaster mapping [49], and the most important, the availability of data in near-real time [34].

There are researching works that study the human behavior in social media [1]; some others study people's feelings (Agarwal et al., 2011). There exists researches that model urban factors analyzing tweets at specific places and developing forecasting processes [77]. In some projects, tweets are considered as social and political indicators [40]; even, there are researches focused on study what motivates people to tweet [58].

On Han et al. (2017) the authors propose a model to study the factors that influence and motivate people to tweet. The researchers established four possible kinds of gratification people get from their participation in Twitter: content, technology, process, and social gratification. Their research shows that users became satisfied

after tweeting relevant information, since they consider themselves useful for other people (social gratification). Burnap and Williams (2016) describe the importance of considering big data and social media as data sources at policy decisions processes, implementing a machine learning classifier to detect cyber hate speech in tweets. Many researches classify human emotions from tweets considering the Ekman and Friesen model, which identifies six basic emotions: joy, anger, fear, sadness, surprise and disgust [25]. Resch et al. (2016) consider such model to identify human emotions implementing a semi-supervised machine-learning algorithm.

Two sciences deeply interested in Twitter data analysis are Geography and Computing. Geography has become indispensable when working with tweets to georeferenced them; computing helps to apply complex processes over tweets to produce data models and statistical studies.

When forecasting and modeling from tweets there are three challenges to consider: i) the spatio-temporal relations between tweets, ii) the temporal evolution of spatially distributed tweets related to specific events; iii) the use of prior geographical knowledge since different geolocations could derive in different events.

In Albuquerque et al. (2015), social media and authoritative data are used to identify information for disaster control. Tweets that provide relevant information about the Elba's river floods in 2013 are analyzed, such data are merged with hydrologic sensors and digital models of the affected areas with a view to create flooding models. Some researches [24, 36, 74] use geographic information systems to georeferenced tweets. This kind of researches are relevant since few tweets have the coordinates of the place where they were posted, as studied in Section 3.1.

One of the main advantages of VGI and social media data are the community ties generated when people share local knowledge with disaster management systems, increasing the survival opportunity for more people [34]. In Landwehr et al. (2016), a social media response system for tsunami warnings is presented, the project is a web application designed for the community of Padang Indonesia, to warn people of tsunamis. The application collects and analyzes tweets from the region to provide immediate feedback when tsunamis happen, supporting an early warning system.

4 Theory. What is the whole point of tweeting? Caution, Users Tweeting

In this approach, we propose a classification for tweets: *the self-centered tweets*, *the social information tweets* and *the collective information tweets* (represented in Figure 1). The individual information or *self-centered tweets*, share personal information of the users answering to the questions “*What am I doing right now?*” or “*What am I seeing right now?*” These tweets are posted quickly and their temporal importance is short depends of the time the user spends doing the action.

Social information tweets express the user's thoughts about social events (political events, strikes, terrorism), which brings about mass mobilization. Opinion mining and sentiment analysis studies are often based on this kind of tweets. Researches as the Resch et al. (2015) make use of this kind of tweets to get information from the people's environment treating Twitter users as sensors.

Collective information tweets, report information about an unusual event occurrence that disturbs the daily life, those tweets are based on individual’s observations but are not self-centered. Figure 2 presents some examples of the proposed classification of tweets.

Tweets provide two valuable data: the user’s perspective of an event, and the coordinates of the place where it is probably happening.

The present analysis is focused on the collective information and the social information tweets. These two kinds of tweets provide information about problems that affect the normal activities of people, or modify their environment, such as natural phenomena or socio-political changes.

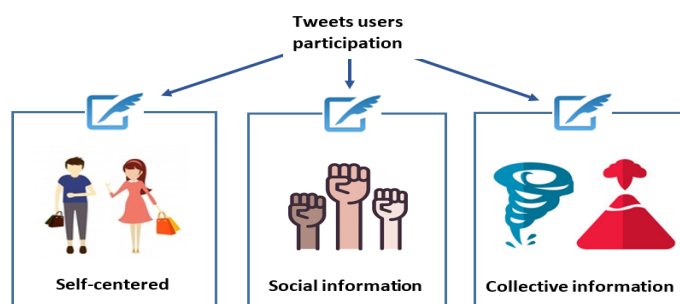


Fig. 1. The tweets can be classified as: 1) self-centered, 2) social information related or 3) collective information related.

4.1 Why are tweets scientifically relevant?

The collective tweets can indeed spread life-saving information. This kind of tweets are frequently collected as VGI for emergency warning and situational awareness, their relevance lies in their facility to spread important information in a short time [65]. When an unusual event occurs, some users near to the place report in Twitter what they see using hash tags and specific words, in some minutes the information is available for millions of users near or not to the event.

Two characteristics of tweets that make them relevant for scientific purposes are their coordinates and timestamp; unfortunately, not all the posted tweets have coordinates, and not all the cities present the same citizen’s participation in the social network.

Geographic science has an important role managing the coordinates of the tweets; furthermore, tweets commonly refer to geographic places on their texts [31]. There exist two possible cases when analyzing tweets coordinates, their presence or absence. Some users let their smartphones add their geographic coordinates when tweeting, in such cases, the Twitter app will automatically enable the device’s embedded GPS to add the latitude and longitude coordinates of the user to the tweets metadata [[17].

When the user tweets from a device without embedded GPS, its geographic coordinates can be manually added; Twitter provides a list of cities based on the users IP address [48]. Those georeferenced tweets however represent a small part (1%-2%) of the whole Twitter stream [8]. The coordinates of the tweet can be represented as point-shaped data using Geographic Information Systems (GIS).

When the user does not allow Twitter to add his coordinates, the tweet can be georeferenced if needed. There have been implemented different kinds of georeferencing models; some of them use specialized gazetteers to estimate the coordinates of the tweet based on its text [38]. The general process consists of three steps: 1.-identification of geographic elements such as roads, avenues, monuments, and buildings names in the text of the tweet; 2.-search the coordinates of the geographic elements in a gazetteer; 3.-compute the coordinates of the tweet, applying geospatial functions over the coordinates found in step 2.

A useful element in tweets is the hashtag (#) used to cluster tweets with similar information [11]. Names and words ambiguity must be considered when using any georeferencing method, to differentiate geographic entities with the same name (as roads, avenues and bridges that share their name), and to prevent the data lack of precision. These issues are addressed by special techniques, which identify road segments and key words from a particular geographic context [5]. One of such techniques is the Named Entity Recognition (NER), which determines the geographic place referenced in a text, considering the disambiguation of names and the most appropriate geographic context (Daly et al., 2013).



Fig. 2. Examples of the three different proposed classes of tweets. The tweets were collected on 21st September 2017, they refer to the earthquake occurred in Mexico on 19th September 2017.

In Roller et al. (2012), NER and Wikipedia are merged to analyze and find geographic terms in tweets. Graham et al. (2014) blend NER and specialized gazetteers to identify geographic sites. Lee et al. (2015) propose a machine learning approach to extract named entities from tweets text and determine their location considering disambiguation.

In Escamilla et al. (2016), is proposed a methodology to geocoding traffic events reported in tweets, the approach identifies the traffic events, geographic features, and their possible spatial relations, using natural language processing (NLP), an ontology, and classification algorithms. In its geocoding stage the proposed methodology extracts geographic entities using NER, finds the spatial relations between them using a syntactic dependency tree, and defines spatial operations to apply over the geographic elements.

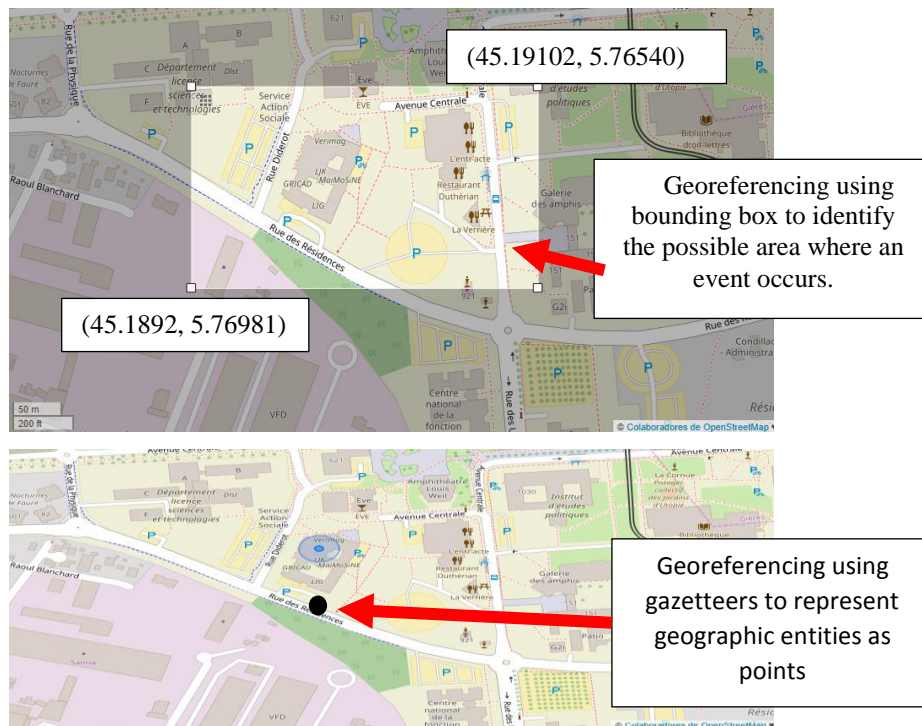


Fig. 3. The two described georeferencing methods when working with tweets. The image on the top shows a bounding box used to search the possible location of a tweet. The bottom image shows a point shape location assigned to a tweet. The cartography belongs to ©OpenStreetMap Contributors.

Another common georeferencing method consist in creating a bounding box; a bounding box is a rectangle created to represent the possible area where a tweet was posted; the box corners are coordinates of places mentioned in the tweets text, or places inferred from other user's tweets [59]. Tweets georeferenced by bounding box are represented as rectangles in GIS.

The geocoding procedures are applied over texts in order to identify geographic entities such as points of interest, events of interest, geographic relations between entities (intersections or their position with respect to others), or geographic features of certain events. Figure 3 shows examples of the mentioned georeferencing methods.

5 Results and Discussions: Geospatial Analysis Based on Tweets

There are researches interested in analyzing tweets to obtain information for problems solving and to develop decision-making processes. Most of the times, the text of the tweets is the main studied feature. Until 2015, a study of 92 revised researching articles from different disciplines working with Twitter showed that only 33% of them made use of the spatio-temporal and semantic characteristics of tweets [67]. Most of the researches interested in Twitter are focused on their text treatment to get information about specific topics, as users emotions [75, 38, 23], or activities. Such researches have propitiated the generation of text mining and natural language processing techniques, and more accurate geo-location procedures.

In this section, some methods to collect and pre-process tweets are explained; also, some common problems present when managing tweets are exposed. Finally, examples of researches that use tweets as data sources are described.

5.1 Preprocessing Tweets

Tweets need to be pre-processed before be used at specific tasks. There is possible to apply over them common data treatment processes, such as text mining, natural language processes, classification, identification of patterns and elements, among others [68].

Twitter lets programmers and developers access to tweets for educational or scientific purposes, there have been developed specialized API's for different programming languages such as Python, R, Java, and others, that allow programmers to extract tweets and its metadata in real time [60].

When the data feature of interest is the text of the tweet, data mining and the NLP are two relevant tools. Commonly, texts are tokenized into words or n-grams to analyze each textual element in order to identify relevant terms and stop words [66]. There exist a variety of algorithms and libraries to implement NLP and text mining procedures over text [78].

When using NLP and textual elements from tweets is important to consider the language used in the region where the tweets were posted [29], there are particularities in the language used by citizens in a region and people who lives in another area, even if the two places share the same official language. For example, in England people use the term motorway to refer to a highway. In addition, it is important to consider the regional terms and their meanings, in Mexico the term *ahorita*, according to the Oxford dictionary [55] means a moment close to the present, immediately after the moment when term is being used. For Mexicans *ahorita* means in a moment or never, according to the context where the word is used.

When working with tweets is important to consider the language in which they are written, in some cases the geographic place where they were tweeted is also relevant. Twitter trending topics change from one geographical area to another; what is more,

people use words that sometimes have a specific local meaning. The miss consideration of the tweets language generates a noisy analysis (Figure 4).

The recurrent task done over the coordinates of tweets is their georeferencing using GIS functions. The spatio-temporal researches that use Twitter, consider three main features of the tweets: their texts, their coordinates and their timestamps [69].

5.2 What are the Main Problems when Working with Data from Tweets?

From the analysis of methodologies that use Twitter as a data source, it is possible to identify two main problems when working with tweets, the data quality and the data quantity they provide.

5.3 Data Quality

Although some tweets are deliberately created to spread useful information within a large community, most of the tweets gathered by scientists for scientific purposes are merely individual information, users are usually not aware their data can be analyzed for scientific purposes. Harvey (2013) proposes the term *contributed data*, to describe any data generated by users' activities on social media platforms. Tweets are part of soft data, which defines all information created and shared on social media platforms; contrary to hard data (traditional datasets managed by governmental agencies), tweets come from heterogeneous sources that cannot be all confirmed.

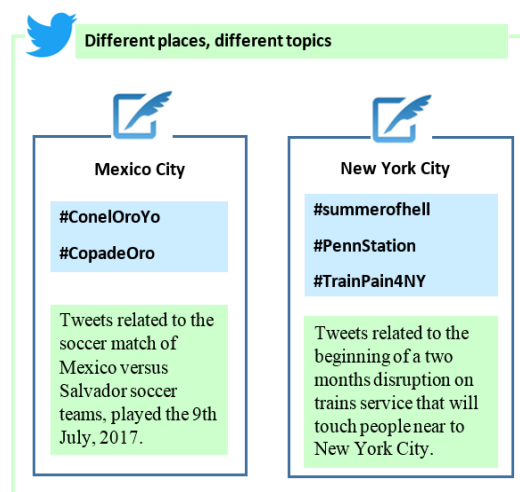


Fig. 4. Trending topics on Twitter at Mexico City and New York City on 10th July 2017. In the first city, the hashtags are associated to a soccer match; in the second case, the hashtags communicate train disruptions. The user's interest change from one geographic area to another.

Moreover, the process of the acquisition is undocumented and there is no data quality assertion [26]. Two recurrent questions are therefore: *Are tweets trustworthy? What is the degree of trust of tweets?*

Tweets quality is therefore heterogeneous and highly variable, while some tweets report a mere observation with no further information (e.g. "It's raining" without any

detail about the rainfall intensity), others may include accurate information (e.g. "Hail cracked the windshield" may provide clues about hail size and intensity). Also, the geo-references given in the text of tweets sometimes are vague or their description is general (e.g. "Earthquake in Mexico" does not point the exact area where the event occurred, meanwhile "Earthquake in Chiapas southwest of Mexico" provides more information to locate the event).

The quality in VGI data is variable for many reasons such as the volunteers experience producing information, their activities and educational level, the multiple technologies to generate information, the purpose of the transmitter, and the precision of the message. Senaratne et al. (2017) review different VGI quality measures and indicators with a view to show the importance of evaluate the VGI information before use it.

In Dashti et al., (2014) the scientists deal with natural hazards using georeferenced tweets and real-time information from a recovery system of disaster information. Crosschecking tweets location with official datasets (flood hazards cartography and Landsat remote imagery), the authors show that georeferenced tweets provide relevant and valuable information about flood events.

Tweets are reports of what users perceive in their local environment; such reports vary from one individual to another, according to their personal life experiences. Observations shared in tweets may be subjective and some bias may be introduced.

Do tweets show people true thoughts? Every time a political event of global interest occurs, *traditional surveys* (based on representative population samples) and *big data analytics algorithms* (opinion mining based on machine learning algorithms that analyse people's publications on Twitter and Facebook) compete to predict the most accurate results.

Which of them got the most relevant results? Opinion mining algorithms based on social media content analysis performed by private companies turned out to be closer to the results than the surveys. The algorithms predicted Trump's election [9] and the Brexit decision [15], but they finally failed in predicting the results of the first French presidential election round [41]. Why? People using social media to express their opinion are not likely to give their true thoughts, or they are ironic, which makes no sense for the machine learning algorithms causing their failure [52]. Finally, users may discuss distant events; users can produce information about a particular event, even if they did not attend it; also, they can keep spreading information after the end of the event. Miller (2017) calls this phenomenon *asynchronous telepresence*.

5.3.1 Data Quality

Tweets are not representative samples of a population, the number of Twitter users is a portion of the total habitants in a specific geographic area (see Figure 5); most of the times people interested in Twitter are part of groups with a specific age and educational level. Tweets can be considered as digital footprints.

Li et al. (2013) tried to draw social profiles of Twitter users in California crossing socio-economic data, demographic data, and geolocated tweets; these results cannot be generalized to the tweeters in the whole world. There is an issue dealing with results generalization, Goodchild (2013) pointed out that results of experiments using random representative samples can be generalized because they apply to the whole

population in a specific area, but the results of a study based on tweets cannot be generalized to a complex geographic and social world. Tweeting indeed involves a minority of the population; tweets show the users concerns but, How can be known if all the people living at the same place share these concerns? Whatever could be concluded from a study case might not show the same results when analyzing the same case in another space or in another period.

Tweets spatial distribution is quite irregular. Studies show that a strong correlation between tweets and populations densities results in high concentrations of tweets in highly populated cities, whereas in rural areas less tweets are produced [48]. That is a reason why the specific geographic areas of study must be delimited when working with Twitter.

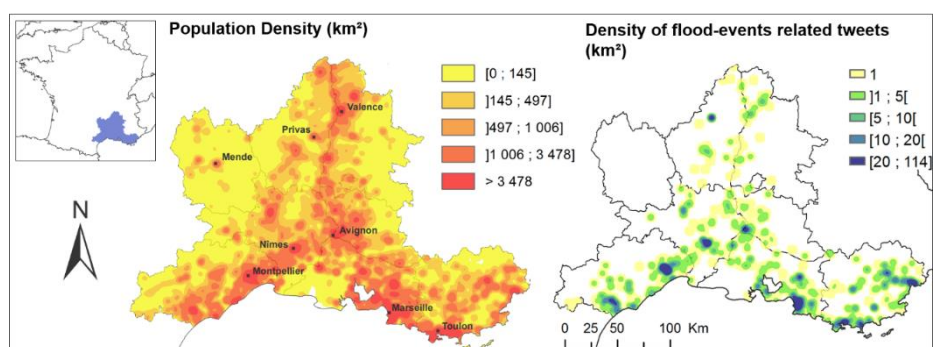


Fig. 5. Analysis of population density and tweets related to floods in the southeast of France. These maps confirm that the number of tweets is not proportional to the population density of an area. Not all the habitants are interested in participating in Twitter.

If events occur in sparsely populated places, just a few users may be therefore involved in creating sparse information. At a local scale, population densities vary depending on time: if an event is occurring within a period of time when a particular place is unoccupied (e.g. a residential neighborhood during daylight), it may be missed and undocumented.

On one hand, events with different features do not have the same impact in the media. Major events, are widely spread in the media and bring about high streams of tweets; on the other hand, local events do not have such a media propagation, so the spatiotemporal extent of the event does not spread over the edges of the physical event. When working with tweets related to events at a local scale, researchers must deal with small datasets.

5.4 Where Twitter Meets GIS

There exist researches that merge GIS and Twitter with a view to develop decision-taking and urban computing procedures. Nowadays some GIS systems let people manage geographic information, even if they do not have a specialized formation [39].

GIS can manage tweets coordinates to represent them as geographic points, or to apply over them geospatial functions; there are open source GIS such as QGIS, Grass, gvSIG, SAGA, and open source databases to manage geographic data such as

PostGIS, SpatialLite and MySQLSpatial [28]. Some GIS-Twitter researches are focused on analyzing people's reports of weather, rain, snow storms, floods [4, 6]; some others model the activities of people in urban environments, such as traffic, pollution, public demonstrations, taxi routes [18, 42, 44]. In addition, there exists some researches interested in studying and managing critical situations [56, 53, 16].

One of the main purposes of Twitter interested researches is to develop analytical tools to study events that represent a risk for people, in order to prevent them from possible damages. In Cavalière et al. (2016), the researches analyze georeferenced tweets posted by users and authorities from the southeast France that provide information about extreme rainfalls and floods. Their proposed methodology processes the tweets using the abductive reasoning. The approach is proved considering tweets posted in the study area during the 2014-fall season.

The Peta Jakarta Project has been implemented to gather crowdsourced data and tweets posted when a flood event occurs. These data are crosschecked with other data sources (topographic, demographic and socio-economic data) and shown in a map, in order to visualize the affected areas and to organize emergency responses [51].

Events forecasting from social media streams is a novelty-researching trend. In Zhao et al. (2016), the authors propose a new batch and online approaches for spatio-temporal events forecasting from social media data; their approach characterizes the evolutionary pattern of both spatial burstiness and structural context. The authors remark the Twitter fundamental characteristics: the timelessness of messages, the ubiquity of social sensors, and the geo-information availability. The researchers conclude that the forecasting of spatiotemporal events requires the consideration of spatial features and their correlations, in addition to the temporal dimension.

Machine learning (ML) algorithms are used to classify and compute regressions over the tweets; for example, in He et al. (2017), the authors propose an optimization framework to extract traffic related information from tweets using a transformation matrix, also make long-term traffic predictions using linear regressions; their study case is San Francisco Bay, California.

An example where GIS and ML are used over tweets is the traffic analysis based on short text from social media [64]. Traffic related tweets are pre-processed by text mining and NLP procedures, georeferenced, and classified by ML algorithms into different traffic events; finally, the classified tweets are geovisualized using an open source GIS, with a view to analyze traffic in Mexico City.

The relevance of Twitter usage in geographic studies relapses in the tweets coordinates and the information that can be inferred from their texts [34]. Tweets are a source of updated reports about what is happening in real world. The analysis of tweets regarding natural hazard lets geography has a perception of the territory changes, to generate updated maps, and to design evacuation routes [46].

When geographic sciences and urban computing use Twitter, they have the opportunity to obtain citizen's information about what happens in their environment, in order to ameliorate their life's modifying urban aspects such as traffic, pollution, and dangerous places. Twitter opens a new opportunity to sense the real world problems, and to implement efficient and fast solutions with the purpose of solve them.

6 Conclusions

Twitter has demonstrated to be a useful tool for data analytics at different science disciplines. Tweets are a new way to obtain information about what happens in real world. People are getting more interested in participating in the social media; they think their information helps others.

Many factors motivate people to tweet; the factors are related to the kind of gratification people gets after do it. Tweets have important features such as metadata and text that can be used to georeference the tweet. Technological advances in mobile devices such as smartphones with embedded GPS let users keep in touch with the social media all the time, and add their coordinates to the tweet metadata. Unfortunately, for scientific analysis based on tweets, few of them have coordinates

In the present approach, tweets can be classified according to the kind of information they provide into three types: self-centered, social, and collective information. The self-centered tweets provide information about the mood and particular activities of the user. The social information tweets are interested in spreading information about particular events that disturbs people's life such as socio-political movements. The collective information tweets main purpose is to provide data about natural hazards and urban circumstances.

Many social researches study the self-centered and social information tweets since they provide information about people's feelings, activities and opinions. Most of times social and collective information tweets are used on researches interested in forecasting, or modelling urban factors and natural phenomena, considering the geographic features of the tweets and their timestamps.

Computing and geographic sciences work together in order to generate data managing procedures to extract tweets, and to analyze the microblogs features for scientific purposes. There have been developed different API's, text mining and NLP procedures to manage the tweets textual component; also, there have been designed and implemented different methodologies to georeference tweets without coordinates in its metadata.

Data obtained from Twitter can be crosschecked with authoritative data and some other information sources to increase its accuracy, and to provide more opportunities of data analysis. Twitter has open an option to monitor the real world dynamics through the user's perspectives; it could be say that Twitter provides first-hand information.

References

1. Aramo-Immonen, H., Kärkkäinen, H., Jussila, J. J., Joel-Edgar, S., Huhtamäki, J.: Visualizing informal learning behavior from conference participants' Twitter data with the Ostinato Model. *Computers in Human Behavior*, 55, 584–595 (2016)
2. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: *Proceedings of the workshop on languages in social media*, pp. 30–38. Association for Computational Linguistics (2011, June)
3. Balazs, J. A., Velásquez, J. D.: Opinion mining and information fusion: a survey. In: *Information Fusion* 27: 95–110 (2016)

4. Basnyat, B., Anam, A., Singh, N., Gangopadhyay, A., Roy, N.: Analyzing Social Media Texts and Images to Assess the Impact of Flash Floods in Cities. In: Smart Computing (SMARTCOMP), 2017 IEEE International Conference on, pp. 1–6 (2017, May)
5. Blanco, R., Ottaviano, G., Meij, E.: Fast and space-efficient entity linking for queries. In: ACM, Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 179–188 (2015, February)
6. Bobb, J. F., Ho, K. K., Yeh, R. W., Harrington, L., Zai, A., Liao, K. P., Dominici, F.: Time-course of cause-specific hospital admissions during snowstorms: an analysis of electronic medical records from major hospitals in Boston, Massachusetts. *American journal of epidemiology* 185(4), 283–294 (2017)
7. Bonney, R., Shirk, J. L., Phillips, T. B., Wiggins, A., Ballard, H. L., Miller-Rushing, A. J., Parrish, J. K.: Next steps for citizen science. *Science* 343(6178), 1436–1437 (2014)
8. Bouillot, F., Poncet, P., Roche, M.: How and why exploit tweet's location information? In: AGILE'2012: 15th International Conference on Geographic Information Science, pp. N–A (2012, August)
9. Breur, T.: US elections: How could predictions be so wrong? (2016)
10. Bruce, E., Albright, L., Sheehan, S., Blewitt, M.: Distribution patterns of migrating humpback whales (*Megaptera novaeangliae*) in Jervis Bay, Australia: A spatial analysis using geographical citizen science data. *Applied Geography*, 54, 83–95 (2014)
11. Bruns, Axel, Jean E. Burgess.: The use of Twitter hashtags in the formation of ad hoc publics. In: Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference 2011 (2011)
12. Burnap, P., Williams, M.L.: Us and them: identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data Science*, 5(1), 11. Earth observation sources: a workflow for volunteered geographic information sensing, Doctoral dissertation, UCL (2016)
13. Burns, R.: Moments of closure in the knowledge politics of digital humanitarianism. *Geoforum*, 53, 51–62 (2014)
14. Cavalière, C., Davoine, P. A., Lutoff, C., Ruin, I.: Analyser des tweets géolocalisés pour explorer les réponses sociales face aux phénomènes météorologiques extrêmes. In: SAGEO'2016 (2016, December)
15. Celli, F., Stepanov, E., Poesio, M., Riccardi, G.: Predicting Brexit: Classifying agreement is better than sentiment and pollsters. In: Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES), pp. 110–118 (2016)
16. Chatfield, A.T., Brajawidagda, U.: Twitter early tsunami warning system: A case study in Indonesia's natural disaster management. *IEEE, System sciences (HICSS)*, 46th Hawaii international conference on, pp. 2050–2060 (2013, January)
17. Cheng, S., Fang, J., Hristidis, V., Madhyastha, H.V., Mithun, N.C., Perkins, D., Tsotras, V.J.: OSNI: Searching for Needles in a Haystack of Social Network Data. In: EDBT, pp. 616–619 (2016)
18. Cottrill, C., Gault, P., Yeboah, G., Nelson, J. D., Anable, J., Budd, T.: Tweeting Transit: An examination of social media strategies for transport information management during a large event. *Transportation Research Part C: Emerging Technologies*, 77, 421–432 (2017)
19. Daly, E.M., Lecue, F., Bicer, V.: Westland row why so slow? fusing social media and linked data sources for understanding real-time traffic conditions. In: ACM, Proceedings of the 2013 international conference on Intelligent user interfaces, pp. 203–212 (2013, March)
20. Danielsen, F., Jensen, P.M., Burgess, N.D., Altamirano, R., Alviola, P.A., Andrianandrasana, H., Enghoff, M.: A multicountry assessment of tropical resource monitoring by local communities. *BioScience* 64(3), 236–251 (2014)
21. Dashti S., Palen L., Heris M., Anderson K, Anderson S., Anderson T.: Supporting disaster reconnaissance with social media data: a design-oriented case study of the 2013 Colorado

- floods. In: Proceedings of the 11th international conference on information systems for crisis response and management. ISCRAM, pp. 630–639 (2014)
22. De Albuquerque, J.P., Herfort, B., Brenning, A., Zipf, A.: A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management. *International Journal of Geographical Information Science* 29(4), 667–689 (2015)
 23. Do, H.J., Lim, C.G., Kim, Y.J., Choi, H.J.: Analyzing emotions in twitter during a crisis: A case study of the 2015 middle east respiratory syndrome outbreak in Korea. In: *Big Data and Smart Computing (BigComp)*, 2016 International Conference on, pp. 415–418. IEEE (2016, January)
 24. Dredze, M., Paul, M.J., Bergsma, S., Tran, H. Carmen: A twitter geolocation system with applications to public health. In: *AAAI workshop on expanding the boundaries of health informatics using AI (HIAI)*, pp. 20–24 (2013, June)
 25. Elfenbein, H.A., Ambady, N.: On the universality and cultural specificity of emotion recognition: a meta-analysis. *Psychological bulletin* 128(2), 203 (2002)
 26. Elwood, S., Goodchild, M.F., Sui, D.: Prospects for VGI research and the emerging fourth paradigm. *Crowdsourcing geographic knowledge*, pp. 361–375 Springer Netherlands (2013)
 27. Escamilla, I., Torres-Ruiz, M., Moreno-Ibarra, M., Quintero, R., Guzmán, G., Luna-Soto, V.: Geocoding tweets approach based on conceptual representations in the context of the knowledge society. *International Journal on Semantic Web and Information Systems (IJSWIS)* 12(1), 44–61 (2016)
 28. Farkas, G.: Applicability of open-source web mapping libraries for building massive Web GIS clients. *Journal of Geographical Systems* 3(19), 273–295 (2017)
 29. Gómez-Adorno, H., Markov, I., Baptista, J., Sidorov, G., Pinto, D.: Discriminating between similar languages using a combination of typed and untyped character n-grams and words. *VarDial* 2017, 137 (2017)
 30. Graham, M., Hale, S.A., Gaffney, D.: Where in the world are you? Geolocation and language identification in Twitter. *The Professional Geographer* 66(4), 568–578 (2014)
 31. Gutierrez, C., Figuerias, P., Oliveira, P., Costa, R., Jardim-Goncalves, R.: Twitter mining for traffic events detection. In: *Science and Information Conference (SAI)*, pp. 371–378. IEEE (2015, July)
 32. Han, Y., Hong, B., Lee, H., Kim, K.: How do we Tweet? The Comparative Analysis of Twitter Usage by Message Types, Devices, and Sources. *The Journal of Social Media in Society* 6(1), 189–219 (2017)
 33. Harvey, F.: To volunteer or to contribute locational information? Towards truth in labeling for crowdsourced geographic information. *Crowdsourcing Geographic Knowledge*, pp. 31–42. Springer Netherlands (2013)
 34. Haworth, B.: Emergency management perspectives on volunteered geographic information: Opportunities, challenges and change. *Computers, Environment and Urban Systems*, 57, 189–198 (2016)
 35. He, J., Shen, W., Divakaruni, P., Wynter, L., Lawrence, R.: Improving Traffic Prediction with Tweet Semantics. In: *IJCAI*, pp. 1387–1393 (2013, August)
 36. Huang, B., Carley, K.M.: On Predicting Geolocation of Tweets using Convolutional Neural Networks. *arXiv preprint arXiv:1704.05146* (2017)
 37. Jiang, B., Thill, J.C.: Volunteered Geographic Information: Towards the establishment of a new paradigm (2015)
 38. Jusupova, Â., Batista, F., Ribeiro, R.: Characterizing the Personality of Twitter Users based on their Timeline Information. In: *ATAS Conferência APSI*, Vol. 16, No. 16, pp. 292–299 (2017, February)

39. Johnson, P.A., Sieber, R.E.: The Geoweb for community-based organizations: Tool development, implementation, and sustainability in an era of Google Maps. *The Journal of Community Informatics* 13(1) (2017)
40. Kavanaugh, A.L., Sheetz, S.D., Sandoval-Almazan, R., Tedesco, J.C., Fox, E.A.: Media use during conflicts: Information seeking and political efficacy during the 2012 Mexican elections. *Government Information Quarterly* 33(3), 595–602 (2016)
41. Kennedy, R., Wojcik, S., Lazer, D.: Improving election prediction internationally. *Science*, 355(6324), 515–520 (2017)
42. Kumar, K.E., Ahmed, H.A.: Estimation of traffic with accuracy through Twitter stream analysis. *International journal of innovative technologies* 4(8), 1317–1324 (2016)
43. Kumar, S., Morstatter, F., Liu, H.: Twitter data analytics. In: *Springer Science and Business Media* (2013)
44. Krueger, R., Sun, G., Beck, F., Liang, R., Ertl, T.: TravelDiff: Visual comparison analytics for massive movement patterns derived from twitter. In: *Pacific Visualization Symposium (PacificVis)*, 2016 IEEE, pp. 176–183 (2016, April)
45. Kryvasheyev, Y., Chen, H., Obradovich, N., Moro, E., Van Hentenryck, P., Fowler, J., Cebrian, M.: Rapid assessment of disaster damage using social media activity. *Science advances*, 2(3), e1500779 (2016)
46. Landwehr, P.M., Wei, W., Kowalchuck, M., Carley, K.M.: Using tweets to support disaster planning, warning and response. *Safety science*, 90, 33–47 (2016)
47. Lee, S., Farag, M., Kanan, T., Fox, E.: A Read between the lines: A Machine Learning Approach for Disambiguating the Geo-location of Tweets. In: *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 273–274. ACM (2015, June)
48. Li, L., Goodchild, M.F., Xu, B.: Spatial, temporal, and socioeconomic patterns in the use of Twitter and Flickr. *Cartography and geographic information science* 40(2), 61–77 (2013)
49. McDougall, K.: Using volunteered information to map the Queensland floods. In: *Proceedings of the 2011 Surveying and Spatial Sciences Conference: Innovation in Action: Working Smarter (SSSC 2011)*, pp. 13–23. Surveying and Spatial Sciences Institute (2011)
50. Miller, H.J.: Time geography and space–time prism. *The International Encyclopedia of Geography* (2017)
51. Meier, P. *Digital humanitarians: how big data is changing the face of humanitarian response*. Crc Press (2015)
52. Mostafa, M.M.: More than words: Social networks’ text mining for consumer brand sentiments. *Expert Systems with Applications* 40(10), 4241–4251 (2013)
53. Ngo, M.Q., Haghighi, P.D., Burstein, F.: A Crowd Monitoring Framework using Emotion Analysis of Social Media for Emergency Management in Mass Gatherings. *arXiv preprint arXiv:1606.00751* (2016)
54. OSM Tasking Manager. Retrieved from <http://tasks.hotosm.org/about>. 16th September (2017)
55. Oxforddictionaries. Spanish Oxford living dictionaries. Retrieved from <https://es.oxforddictionaries.com/translate/spanish-english/ahorita?locale=en>. 16th September (2017)
56. Panagiotopoulos, P., Barnett, J., Bigdeli, A.Z., Sams, S.: Social media in emergency management: Twitter as a tool for communicating risks to the public. *Technological Forecasting and Social Change*, 111, 86–96 (2016)
57. Park, H., Reber, B.H., Chon, M.G.: Tweeting as health communication: health organizations’ use of twitter for health promotion and public engagement. *Journal of health communication* 21(2), 188–198 (2016)
58. Pentina, I., Basmanova, O., Zhang, L.: A cross-national study of Twitter users’ motivations and continuance intentions. *Journal of Marketing Communications* 22(1), 36–55 (2016)

59. Pereira, J., Pasquali, A., Saleiro, P., Rossetti, R.: Transportation in Social Media: an automatic classifier for travel-related tweets. arXiv preprint arXiv:1706.05090 (2017)
60. Perera, R.D., Anand, S., Subbalakshmi, K.P., Chandramouli, R.: Twitter analytics: Architecture, tools and analysis. In: Military Communications Conference, 2010-MILCOM 2010, pp. 2186–2191. IEEE. (2010, October)
61. Resch, B., Summa, A., Zeile, P., Strube, M.: Citizen-Centric Urban Plannin through Extracting Emotion Information from Twitter in an Interdisciplinary Space-Time-Linguistics Algorithm. *Urban Planning* 1(2), 114–127 (2016)
62. Ribeiro Jr, S.S., Davis Jr, C.A., Oliveira, D.R.R., Meira Jr, W., Gonçalves, T.S.: Pappa, G.L.: Traffic observatory: a system to detect and locate traffic events and conditions using Twitter. In: Proceedings of the 5th ACM SIGSPATIAL International Workshop on Location-Based Social Networks, pp. 5–11. ACM (2012, November)
63. Roller, S., Speriosu, M., Rallapalli, S., Wing, B., Baldrige, J.: Supervised text-based geolocation using language models on an adaptive grid. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1500–1510. Association for Computational Linguistics (2012, July)
64. Saldana-Perez, A.M.M., Moreno-Ibarra, M.: Traffic analysis based on short texts from social media. *International Journal of Knowledge Society Research (IJKSR)* 7(1), 63–79 (2016)
65. Sheppard, S.A.: wq: A modular framework for collecting, storing, and utilizing experiential VGI. In: Proceedings of the 1st acm sigspatial international workshop on crowdsourced and volunteered geographic information, pp. 62–69. ACM (2012, November)
66. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic n-grams as machine learning features for natural language processing. In: *Expert Systems with Applications* 41(3), 853–860 (2014)
67. Steiger, E., Albuquerque, J.P., Zipf, A.: An advanced systematic literature review on spatiotemporal analyses of Twitter data. *Transactions in GIS* 19(6), 809–834 (2015)
68. Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In: *Social computing (socialcom), 2010 IEEE second international conference on*, pp. 177–184, IEEE (2010, August)
69. Summa, A., Resch, B., GIS, G.Z., Strube, M.: Microblog emotion classification by computing similarity in text, time, and space. In: *Proceedings of the Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*, pp. 153–162 (2016, December)
70. Taylor, M., Wells, G., Howell, G.: Raphael, B.: The role of social media as psychological first aid as a support to community resilience building. *Australian Journal of Emergency Management* 27(1), 20 (2012)
71. Van Dijck, J.: *The culture of connectivity: A critical history of social media*. Oxford University Press (2013)
72. Wang, Q., Bhandal, J., Huang, S., Luo, B.: Classification of Private Tweets Using Tweet Content. In: *Semantic Computing (ICSC)*, In: 2017 IEEE 11th International Conference on, pp. 65–68. IEEE (2017, January)
73. Whittaker, J., McLennan, B., Handmer, J.: A review of informal volunteerism in emergencies and disasters: Definition, opportunities and challenges. *International journal of disaster risk reduction*, 13, 358–368 (2015)
74. Williams, E., Gray, J., Dixon, B.: Improving geolocation of social media posts. *Pervasive and Mobile Computing*, 36, 68–79 (2017)
75. Wu, X., Bartram, L., Shaw, C.: Plexus: An Interactive Visualization Tool for Analyzing Public Emotions from Twitter Data. arXiv preprint arXiv:1701.06270 (2017)

76. Zhao, L., Chen, F., Lu, C.T., Ramakrishnan, N.: Dynamic theme tracking in Twitter. In: Big Data (Big Data), 2015 IEEE International Conference on, pp. 561–570. IEEE (2015, October)
77. Zhao, L., Chen, F., Lu, C.T.: Ramakrishnan, N.: Online Spatial Event Forecasting in Microblogs. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 2(4), 15 (2016)
78. Zhou, Y., Zhang, Z., Lan, M.: ECNU at SemEval-2016 Task 4: An Empirical Investigation of Traditional NLP Features and Word Embedding Features for Sentence-level and Topic-level Sentiment Analysis in Twitter. In *SemEval@ NAACL-HLT*, pp. 256–261 (2016)

Exploring Storing Capacity of Hyperdimensional Binary Vectors

Job Isaias Quiroz Mercado, Ricardo Barrón Fernández,
Marco Antonio Ramírez Salinas

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación,
Mexico City, Mexico

jobquiroz@hotmail.com, barron2131@gmail.com, marco.a.ramirez.s@gmail.com

Abstract. Hyperdimensional computing is an emergent model of computation based on the manipulation of high-dimensional vectors which are used not only to represent variables and values, but also to represent complex structures such as relations, sets and sequences. All vectors in the model are always the same size either if they represent a single concept or a sequence of objects. Hyperdimensional computing uses reduced representations, since there is a compression process to encode complex structures while maintaining the same size on the output vector. In this paper we explore the storing capacity of hyperdimensional vectors that encode semantic feature norms. We describe a method for encoding and retrieving feature information of concrete concepts and present experimental results of the successful retrieval of such features.

Keywords: hyperdimensional computing, vector symbolic architectures, reduced representations, semantic pointer architecture.

1 Introduction

Hyperdimensional computing (HD computing) is based on the properties of high-dimensional vectors and arithmetical operations perform on them. HD computing takes ideas from artificial neural networks in the sense that processing is performed in a distributed fashion; it is also inspired on symbolic computing because complex structures, such as hierarchical trees or sequences, can be formed by manipulating symbols (vectors) that represent simpler objects. Additionally, HD computing also “includes ideas from probability theory, statistics and abstract algebra” [14].

The use of high-dimensional vectors comes implicit with Artificial Neural Networks (ANN), but in HD computing vectors are not only part of the architecture but are the basic computing entities itself. Hyperdimensional vectors can be manipulated to make associations, form hierarchies and perform other types of cognitive computations, formation of these types of complex structures is one of the weak points of ANN [4].

The main property of HD computing is that it can operate with approximate patterns, providing flexibility to the computing system and allowing it to scale to large learning applications. HD computing is becoming more relevant as a cognitive modeling tool.

Semantic features “represent the basic conceptual components of meaning for any lexical item” [5]. Any lexical term is associated with a set of semantic features each of which constitutes one component of a word’s intension, semantic features try to establish the meaning of a word in terms of its relationships with other words. Since semantic feature are subjective they must be obtained empirically. In [11] McRae et al, describe a set of semantic feature norms collected from approximately 725 participants for 541 concrete concepts.

In this work, we explore the storing capacity of hyperdimensional binary vectors that encode semantic features norms from McRae’s dataset. Our aim is to measure the maximum number of vectors that can be stored, and later retrieved, within a single vector, our hypothesis is that for straightforward encoding methods it is not necessary to have vectors above 5,000 dimensions.

Our experimental results will be used for selecting the appropriate dimensionality of vectors within a HD computing system still in development.

The rest of the paper is organized as follows: Section 2 summarizes several related works to HD computing for concept representation. Section 3 explains the general properties of HD computing and describes how to represent semantic features. In Section 4 we present the experimental results and finally, Section 5 draws the conclusions and future work.

2 Related Work

2.1 Hyperdimensional Computing

In [9] Kanerva introduces the term Hyperdimensional Computing as a model for cognitive computing, in his work he summarizes the main properties of the model, and some current applications for it. But even before that, during the mid-1990s, a class of connectionist network architectures called Vector Symbolic Architectures (VSA) started to be developed. These architectures use arithmetical operations on vectors for encoding structure using distributes representations [7].

The Semantic Pointer Architecture is a VSA developed to model a high-level cognitive system that is also biologically plausible. The main hypothesis for the model is that “Higher-level cognitive functions in biological systems are made possible by semantic pointers.” [1]. A semantic pointer is a high dimensional vector that can be used to access large amounts of information within memory, it has the function of a pointer, but unlike conventional pointers, semantic pointers are similar to the information they point to.

The Semantic Pointer Architecture (SPA) has extensively been in [1], and it has been used in “Spaun”, a large-scale spiking neuron model capable to integrate perception, cognition and action across several different tasks [2]. SPA provide a unified framework to study cognition, to model working memory and to encode natural language sentences, all with the same representation mechanisms: hyperdimensional vectors.

Crawford et al. implemented a large-scale knowledge base called WordNet using the SPA. WordNet is a manually constructed lexical database of the English language [3],

where each word is associated with multiple meanings forming a complex hierarchical structure of concepts. This work is relevant to Hyperdimensional Computing because it demonstrates how hyperdimensional vectors can be used to encode complex cognitive structures while maintaining flexibility and efficiency.

2.2 Semantic Feature Norms

Semantic feature norms are another way to describe a concept, that unlike WordNet definitions, are empirically obtained by interviewing people: each person is presented with a set of concept names and are asked to list the features he thinks “are important for each concept”.

These feature representations have been used to “provide insight into a number of phenomena involving semantic memory and categorization”. Additionally, semantic feature norms can be useful to develop models of semantic memory and concept representation. Since semantic feature norms are directly obtained by asking people they tend to be ambiguous, however this is the type of communication that prevails in the real world.

In this work we use binary high dimensional vectors for encoding concepts described by semantic feature norms from the largest dataset in literature, we explore the capacity for storing and later retrieving each of the features composing a concept description.

3 Hyperdimensional Computing Background

One of the most relevant properties of the high-dimensional spaces is that most of the space is nearly orthogonal to any given point. This means that if two random vectors are generated, it is highly probable (more than 99.999%) they are mutually orthogonal, new symbols (vectors) can be stored into memory without clashing with preexisting elements.

These properties were exploited in the SDM model developed by Kanerva in 1988 [10]. However, the properties of high-dimensional spaces can also be used to perform other type computations, for which is necessary to define a set of HD computing operators.

3.1 HD Computing Operations

HD computing is mainly about manipulating and comparing patterns, such patterns are stored in an associative memory where all original vectors are stored and where new generated vectors can be cleaned up and be approximated to one or more of the original vectors.

HD computing has three main operations: addition, multiplication and permutation. In this work we use binary vectors, addition is an element-wise binary average, multiplication is an element-wise exclusive-or and permutation is realized by a logical shift. A more detailed explanation of HD Computing operations can be found in work of Kanerva [13].

All previous operators allow us to encode, map and retrieve hyperdimensional patterns, but in most cases the retrieval is not going to be exact. For example, $X = X_1 * A + X_2 * B$ is storing the association of A with X_1 and B with X_2 . In order to retrieve A we can multiply X by X_1 :

$$X * X_1 = (X_1 * A + X_2 * B) * X_1 \Rightarrow X * X_1 = A + X_1 * X_2 * B. \quad (4)$$

The resulting vector contains the sum of the desire value (A) and an unknown vector ($X_1 * X_2 * B$), to discriminate this last vector, we can use a clean-up memory that approximates $X * X_1$ to A :

$$Read(A + X_1 * X_2 * B) = Read(A + noise) = A. \quad (5)$$

When a noisy version of an item is given as input, the memory must either output the most similar item, according to a distance metric, in this case hamming distance, or indicate that the input is not close enough to any of the store items.

3.2 Encoding Feature Representations

Feature representation of a concept consists of a list of features that, according to people, are the most important to that concept. A concept feature usually consists of two or more words, usually a verb followed by another concept, depending on these words a feature can be classified within a feature type. In Table 1, we show an example for the feature representation for *knife* found in McRae's dataset.

Table 1. Feature representation for *knife*.

Concept name	Feature	Production frequency	Brain Region Classification
Knife	has a handle	14	Visual-form and surface
	made of steel	8	Visual-form and surface
	is shiny	5	Visual-form and surface
	used for cutting	25	Function
	used for killing	7	Function
	is sharp	29	Tactile
	is dangerous	14	Encyclopedic
	found in kitchens	8	Encyclopedic
	a weapon	11	Taxonomic
	a utensil	9	Taxonomic

The most straightforward way to encode feature representation is by generating two vectors, one for the name of the feature, i.e. 'has', and another for its value, i.e. 'handle', and then using the multiplication operator for binding them together. Once each feature is represented as a pair of binded vectors the next step is to add all the features to produce a single vector that encodes all the features associated with the concept. By applying the proper operations, we can extract the value for each of the features encoded in the vector:

$$knife = has * blade + made.of * steel + texture * shiny + \dots + is * utensil. \quad (6)$$

In the McRae's dataset all concepts have a different number and type of features, and in most cases, there are two or more features of the same type, either because an object is composed by several parts and therefore the 'has' feature type appears more than once, or because an object has several options for some features, i. e. an apple can be red, green or yellow. These cases do not represent a problem in the encoding process because hyperdimensional multiplication distributes over addition [9] and therefore we can encode each feature independently without knowing if there are more features that share the same type:

$$\begin{aligned} \text{apple} &= \text{is} * \text{fruit} + \text{has} * \text{skin} + \text{color} * (\text{red} + \text{green} + \text{yellow}) \\ &= \text{is} * \text{fruit} + \text{has} * \text{skin} + \text{color} * \text{red} + \text{color} * \text{green} + \text{color} * \text{yellow}. \end{aligned} \quad (7)$$

From equation 7 we can see the semantic pointer *apple*, it is a pointer because is a mean to access more information in memory, we can extract each feature value and, since a feature value can be a semantic pointer itself, we can proceed to explore them in more detail. Semantic Pointers can be used to represent hierarchies, Figure 1, where every level in the hierarchy is a deeper exploration within a semantic pointer.

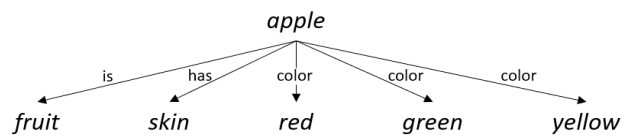


Fig. 1. Semantic Pointer Vectors can represent hierarchical structures.

3.3 Retrieving encoded feature values

Once a concept has been encoded, following equation 6, we have to be able to retrieve the feature values associated with it, to do this the inverse operation has to be applied to the semantic pointer, in the case of binary vectors the XOR operation is its own inverse.

Since in our approach we can have more than one feature value associated to a feature type our clean-up memory outputs, not only the closest vector to the feature value obtained, but all the vectors that are similar enough according to certain threshold value:

$$\text{is} * \text{apple} \cong \text{fruit} \quad \text{color} * \text{apple} \cong (\text{red}, \text{green}, \text{yellow}). \quad (8)$$

To extract all feature values from a semantic pointer it is necessary to know its correspondent feature types, however once a vector is already encoded there is no way to know which feature types are present within this semantic pointer. In order to extract the feature values, we multiply the semantic pointer with all the feature type vectors in the system and, by using the clean-up memory, verify which vectors are meaningful after the multiplication is done. This process is explained in Algorithm 1.

Algorithm 1. Extracting feature value vectors from a semantic pointer.

```
Function GetFeatureValues (SPvector):  
  FeatTypes = [is, has, color, shape, ...] -- All feature types  
  FeatValues = []  
  For each FtType in FeatTypes:  
    ValueVec = FtType * SPvector  
    ValueVec = CleanUpMemory (ValueVec)  
    If ValueVec is not null:  
      ValueVec.push(FeatValeus)  
  Return ValueVec
```

The function CleanUpMemory returns a list of all the FeatureValue vectors that are close enough to the input vector. Two vectors have a very high probability of being the same when its normalized hamming distance is around 0.45.

4 Experimental Results

We performed several experiments to test the maximum capacity for storing and retrieving feature values from semantic pointer vectors representing definitions of concepts from the McRae's dataset.

During the experiment we randomly select a set of concepts from the dataset and encode them into our associative memory, using binary hyperdimensional vectors. Additionally, we store a list of all the feature values for each selected concept to later compare them with the values extracted from the semantic pointers.

The average number of feature values for the concepts in the McRae dataset is around 17 and the maximum being 22, however, to explore the maximum storing capacity of the semantic vectors we defined additional concepts by randomly selecting features from the dataset, the maximum number of features for a concept was 70.

Table 2 shows the percentage of successful retrievals of feature values for N-size semantic pointers and the number of features in vector ranging from 10 to 70. Figure 2 illustrates these same results.

Based on the obtained results and considering that the maximum amount of features in the McRae dataset is 22, we can state that a binary vector of size $N = 2,500$ is enough to store the feature representations of concepts from the used dataset. Even though some sources [8, 9] indicate that 10,000 bits can be convenient for most applications, in our case the size of our vectors might be smaller which has a direct impact on the processing time of our system.

We can also observe how the percentage of retrieved features increases as the dimensionality of the vector increases but not in a linear fashion, for example for a vector size of $N = 5,000$ we can retrieve 86% of all values in a semantic pointer composed by 50 features, if we double the size of the vector, $N = 10,000$, we can retrieve 82% of the features.

Even though this might seem a small increment it is very relevant to our work because, by the way in which we encode all the features, there is no way to know if the

retrieved features are the most important, in terms of frequency for example, for the concept. There might be ways to ensure that the most important features are retrieved first, but for the characteristics of this dataset it is unnecessary.

Table 2. Storing capacity of binary semantic pointers.

Number of Features in Vector	Successfully retrieved features (%)				
	N = 1,000	N = 2,500	N = 5,000	N = 7,500	N = 10,000
10	100 %	100 %	100 %	100 %	100 %
20	95 %	100 %	100 %	100 %	100 %
30	90 %	100 %	100 %	100 %	100 %
40	72.5 %	92.5 %	95.0 %	100 %	100 %
50	70.0 %	68.0 %	86.0 %	92.0 %	82.0 %
60	58.3 %	46.7 %	61.7 %	56.7 %	61.7 %
70	38.6 %	42.9 %	34.3 %	34.3 %	27.1 %

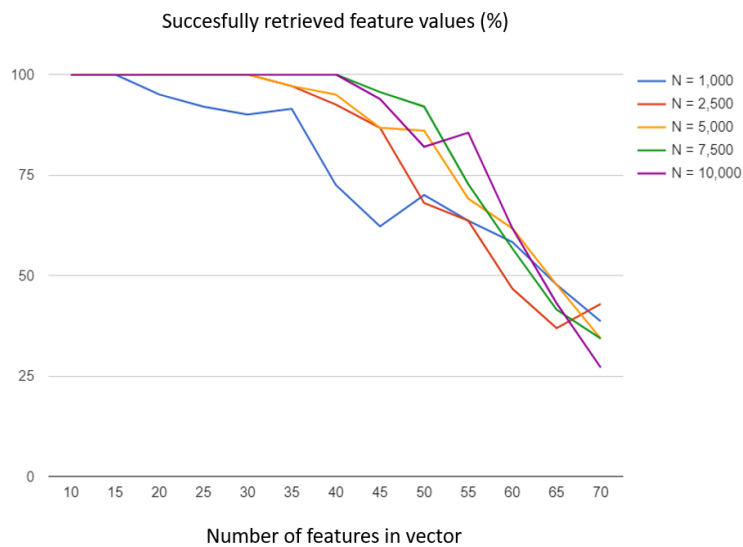


Fig. 2. Storing capacity for N-size semantic pointers.

5 Conclusions and Future Work

This work presents an empirical exploration of the storing capacity of binary semantic pointers as a mean to represent semantic feature norms. We presented some highlights of Hyperdimensional Computing, an emergent model of computation based on the

manipulation of high-dimensional vectors. This type of computation allows us to encode from single concepts up to more complex data structures such as sequences.

We describe how feature representations might be encoded into high-dimensional vectors and we describe a straightforward method to retrieve such vectors. We perform experiments for encoding randomly chosen vectors from the largest known dataset of semantic feature norms.

This work presents preliminary results from a larger project that pretends to encode the entire McRae dataset into a Vector Symbolic Architecture for a Semantic Network, the results presented allow us to decide the size of the vectors to be used in the complete system.

Acknowledgements. This work has been funded by SIP-IPN under grant SIP-20181698 and also by CONACYT scholarship number 666415.

References

1. Eliasmith, C.: *How to build a brain: A neural architecture for biological cognition*. USA: Oxford University Press (2013)
2. Eliasmith, C., Stewart, T., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., *et al.*: A large-scale model of the functioning brain. *Science*, 338 (6111), 1202–1205 (2012)
3. Fellbaum, C.: *WordNet and wordnets*. *Encyclopedia of Language and Linguistics*. Second Edition, Oxford: Elsevier, pp. 665–670 (2005)
4. Fodor, J. A., Pylyshyn, Z. W.: *Connectionism and Cognitive Architecture: A Critical Analysis*. *Cognition* 28(1):3–71 (1988)
5. Fromkin, V., Rodman, R., Hyams, N.: *An Introduction to Language*. Boston, MA: Wadsworth, Cengage Learning, p. 578 (2014)
6. Gallant, S., Okaywe T.: *Representing Objects, Relations and Sequences*. *Neural Computation* 25(8):2038–2078 (2013)
7. Gayler, R.: *Vector Symbolic Architectures answer Jackendoff’s challenges for Cognitive Neuroscience*. In: *ICCS/ASCS International Conference on Cognitive Science*. CogPrints, Sydney, Australia, University of New South Wales, 133–138 (2003)
8. Kanerva, P.: *Computing with 10,000-bit words*. In: *52nd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, pp. 304–3010 (2014)
9. Kanerva, P.: *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High Dimensional Random Vectors*. *Cognitive Computation* 1(2), 139–159 (2009)
10. Kanerva, P.: *Sparse Distributed Memory*. Cambridge, MA: Bradford/MIT Press (1988)
11. McRae, K., Cree, G., Seidenberg, M., McNorgan, C.: *Semantic feature production norms for a large set of living and nonliving things*. *Behavior Research Methods, Instruments & Computers*, 37(4):547–559 (2005)
12. Plate, T.: *Holographic reduced representation: distributed representation of cognitive structure*. Stanford: CSLI (2003)
13. Quiroz, K., Barrón, R., Ramírez, M.: *Sequence Prediction with Hyperdimensional Computing*. *Research in Computing Science*, 138 (2017)
14. Rahimi, A., Datta, S., Kleyko, D., Paxon, E., Olshausen, B., Kanerva, P., Rabaey, J.: *High-Dimensional Computing as a Nanoscalable Paradigm*. *IEEE Transactions on Circuits and Systems: Regular Papers* (99), 1–14 (2017)

Context Pattern Based Agricultural Named Entity Recognition

Payal Biswas, Aditi Sharan, Ashish Kumar

Jawaharlal Nehru University, New Delhi, India
payal.biswas138@gmail.com

Abstract. Named entity recognition (NER) play a vital role in various application of Natural Language processing. Although a significant work has been done in general and biomedical domain NER, but agriculture domain has been ignored for a long time. Agriculture entity includes name of crops, crop diseases, fertilizers etc. Due to the inapplicability of conventional features which has been used for identifying general named entities, recognizing and extracting the agricultural entities become a rigorous and challenging task. As NER in agriculture domain has not been yet explored a lot, thus building up a NER system for agriculture domain is very recent and vital work. This paper proposes a novel context-based approach to develop a NER system for agriculture domain. The proposed approach employs the context pattern for extracting the required entity of interest. The experiment is carried out in two different genres 1) Word Context Pattern 2) POS context pattern. In word context pattern, merely the co-occurring word tokens corresponding to the required entity is considered. While in Part of Speech (POS) rather than considering the co-occurring word tokens, their POS structure is plied. We have proposed seven part of speech patterns which are most likely to comprise all the instances of required entity of interest. The remarkable point is that the proposed POS patterns have not only device the known agricultural entities but have also extracted out 55 hidden entities from the data set. To boost up the performance of the NER system semantic similarity module has also been exercised. The proposed approach attains an accuracy of 70.45 % and recall of 91.3% which is appreciable as the preparatory work.

Keywords: named entity recognition, agriculture NER, word context pattern, POS context pattern, semantic similarity, agriculture entity.

1 Introduction

Named Entity Recognition (NER) has grown as an important area of research in past two decades. It is a fundamental and key component in the field of text mining and Natural Language Processing (NLP). Lisa F. Rau [23] presented the first research paper in this area in 1991 at Seventh IEEE Conference on Artificial Intelligence Application [23] and then after in 1996, after the MUC-6 (Message Understanding Conference-6), it has been accelerated and never been declined since then [9]. In the taxonomy of computational linguistics, it falls under the domain of information extraction. Named Entity recognition can be defined as identifying the references to specific entities like names, including person, organization and location names and numeric expressions including time, date, money, and percent expression [20]. It involves processing structured as well as unstructured text document and extracts these information units

from the text. As named entities are the basic building block of content of the document, extracting these phrases from the document play an important role towards more intelligent information extraction and management. In the preparatory stage the task of NER involves extracting the name of place person and organization etc. in the text. Later on, these categories have been sub-categorize into more fine-grained classes like name of city, state [8, 14], name of film and scientist [7] and so on. The task of NER has also disseminated for different languages like Turkish [28], Arabic [1] and many more. Researchers working over the field of NLP and text mining have also glance their eye over different genre and domain like tweets data [25], clinical data [13], and biomedical data [32]. However, NER for agriculture domain has been ignored for a long time.

As we know India is principally an agriculture-based country and it is the backbone of Indian economy. It contributes a significant figure, approx. 13.7 % * to the Gross Domestic Product (GDP). For last six decades, various government and non-government organizations such as IARI, ICAR, IASRI and FAO are working in the field of agriculture. As a result of this ample amount of digital information become available in the internet. Agriculture domain is very much similar to the biomedical domain, but due to some unfavorable circumstances or unavailable resources no significant work has been done in this field. However, if someone want to do any kind of Natural Language Processing like Information retrieval, Machine translation, document summarization, Semantic web search, Question answering System or any other in agriculture domain one has to find out the agricultural named entities from the document.

Although there are various Named Entity Recognizers available now a day such as Stanford's Named Entity Recognizer, Python NLTK Named Entity Recognizer, Learning Based Java1 (LBJ) [26] and many others. But since they are open domain NER, they can only able to recognize the name of Place, Person and Organization but unable to tag the Agricultural related entities. Similarly, various NER systems like ABNER, BNER, BANNER has been developed for biomedical domain but they are specially trained for recognizing biomedical entities hence they do not perform well for recognizing agricultural entities.

Hence if we want to develop any kind of NLP application for agriculture domain, we have to extract the basic entities particularly the agricultural named entities like plants name, plant diseases and fertilizer's name etc. from the text. A propaedeutic step has been taken by Biswas *et al.* [2] in this field. However, they have suggested solely a framework called AGNER, for building up a NER system for agriculture domain. Although there are various challenges comprising in building up a NER system for agricultural domain, in this paper we have taken an attempt to develop an agricultural NER using context pattern and semantic similarity measure.

2 Challenges with Agriculture NER

Since the research over NER lends its wings, a considerable amount of work has been done in the field of NER. In general, there are two main approaches for developing an NER system: Rule Based Approach and Machine learning approach. However, the rule-based approaches give good results, but developing such rules are time consuming,

importable and require expert's knowledge. Nowadays machine learning techniques for developing the NER systems are in trend. Various machine learning approaches such as HMM [6, 35] SVM [30], CRF, [19, 29] maximum entropy [27] has been employed for framing the NER systems both for open domain NER as well as domain specific NER like biomedical [34, 12, 16] and tweeter [17,25]. These machine learning approaches are applied over some specific features. For an instance orthographic feature, word shape feature, prefix and suffix feature [29] and many others. Orthographic feature describes how a token is structured like ALLCAPS, INITCAP, alpha numeric, word containing roman digit etc. Word shape feature expounds that same category of tokens may share similar shape and prefixes and suffixes are some special sub words with which same class of tokens starts or ends. These features play an evincive role in exercising varied machine learning algorithms. However, the most hitching thing is that these features are not applicable with the agriculture entities. For example, the crop names like rice, wheat, oat, barley etc. neither follow any orthographic feature, nor word shape or prefix, suffix feature. Hence exploiting the machine learning algorithms for extracting the agricultural entities is not facile.

Even though agriculture domain seems similar to biomedical domain however their naming conventions are not same. Unlike agriculture entities, biomedical entities have some special characteristics such as long and descriptive naming sequences, conjunctive and disjunctive structure, abbreviations, cascaded construction etc. Moreover, in case of biomedical domain the presence of two annotated corpora GENIA and GENETAG [33] and the impendence of MEDLINE [31], scientific biomedical knowledgebase aids in fabricating the biomedical NER with satisfactory results. However as per our knowledge, very few works have been done in the field of NER in agriculture domain [2, 4, 18] and there is no such annotated agricultural corpora and full fleshed agricultural knowledge base.

3 Proposed Work

As it is discussed in the previous section that the various word level features like orthographic feature, word shape feature, prefix and suffix feature are not congruent for the agricultural entities hence we have to move for the context-based feature. In this work we have proposed an approach for developing an agricultural named entity recognizer, AGNER, which exploits the context of required named entities. The experiment is carried out in two phases: 1) using the context pattern of words 2) using the context pattern of POS.

3.1 Experiment 1: Word Context Pattern

The context pattern of words is devised in two steps. First step quest for the co-occurring word patterns and the next step check for the accuracy of their co-occurrence.

Co-Occurring Word Pattern Extraction. In order to extract the context patterns of words corresponding to the named entity terms, first of all the required entities are searched in the training dataset. In association with these entities different size context windows are extracted. Context windows are the set of tokens of fix size neighboring

words in the text. Table 1 shows the context window of different size around the token W_0 . In this work the maximum window size is taken as ± 5 .

Table 1. Context window of words.

Window	Description
$W_0 W_{+1}$	A word right to the required token
$W_{-1} W_0$	A word left to the required token
$W_{-1} W_0 W_{+1}$	A word both left and right to the required token
$W_0 W_{+1} W_{+2}$	Two words right to the required token
$W_{-2} W_{-1} W_0$	Two words left to the required token
$W_{-2} W_{-1} W_0 W_{+1} W_{+2}$	Two words left and two words right to the required token
$\dots W_{-2} W_{-1} W_0 W_{+1}$
$W_{+2} \dots$	

As a result of this multitudinous set of word phrases of different window size has been obtained. These word phrases are sorted on the basis of their frequency and few most frequent words patterns are extracted out.

Co-occurrence Evaluation. This step checks for the verity of co-occurrence of extracted terms with the required entity of interest. For this, first of all the terms which have been extracted in the word pattern extraction phase, as a co-occurred term with the required named entity, are retraced in the testing dataset. Now the co-occurrence of these entities has been cross checked with the other entities in the dataset. At last the entities which have been extracted from the probable position of named entity are examined for the required entity of interest. The algorithm, shown in Table 2, presents the implementation detail of the proposed word pattern approach.

Table 2. Algorithm for word context pattern.

Algorithm: Word Context Pattern

Input: D_{TR} – Training Dataset, D_{TS} – Testing Dataset, S – Seed Word

Output: Agriculture Entities $E = \Phi$

Word Pattern Extraction

1. $E \leftarrow \Phi, T \leftarrow \Phi;$
2. for each $s \in S$ do
3. Find context window for W_i in D_{TR} where $i = \pm 5$
4. Extract co-occurred words, $CW_d(i) \leftarrow W_i$
5. Co-occurred words patterns $CW_dP \leftarrow \text{Frequent } CW_d(i)$
6. Word Pattern $P_w = CW_dP - s$

Co-occurrence Evaluation

Algorithm: Word Context Pattern

7. For each window W_i in D_{TS} where $i = \pm 5$
8. Extract n-gram, $N_i \leftarrow W_i$
9. Search for n-gram N , where $N_i := P_w$
10. Extract token T from N , where T is in probable position in N .
11. Agriculture Entity, $E := E + T$

The flow diagram of the process, which is involved in the experiment 1, is shown in Fig 1.

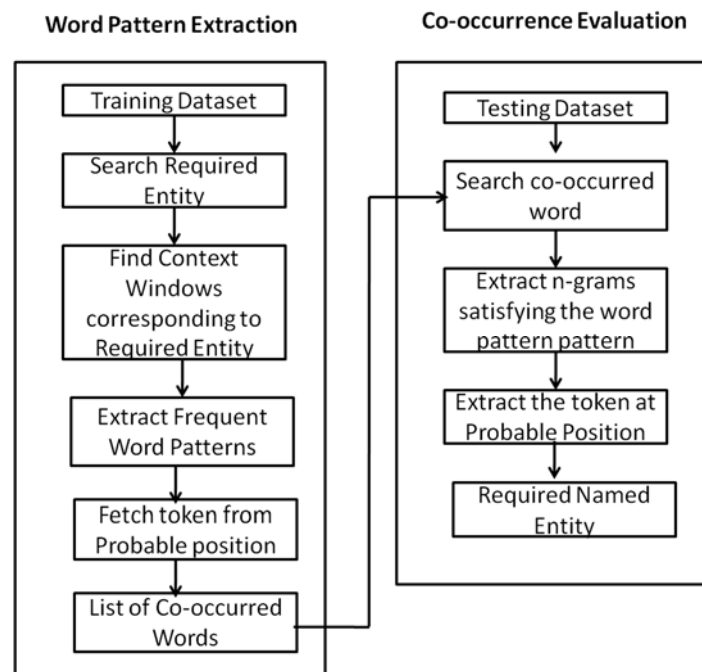


Fig.1. Flow diagram of word context pattern.

3.2 Experiment 2: Part of Speech Context Pattern

Unlike word pattern, in part of speech (POS) context pattern, rather than considering the absolute word, their part of speech structure is captured. This approach works in three steps: Pattern Extraction, Pattern Evaluation, and Pruning.

Pattern Extraction. The word phrases which have been devised from former approach are diverse in nature. Hence in order to achieve the homogeneity among the various word tokens of different size windows these word phrases have parsed for their POS

structure [2]. Once after the obtained word phrases have been scanned for the POS structured, a finite set of unique POS patterns have been emerged. These patterns are then sorted on the basis of their frequency and few of the most frequent POS patterns are extracted. As a consequence, seven most frequent patterns have been derived. Table 3 quotes varied POS patterns, which has been instated from the context word window. In these patterns NN stands for noun phrase, IN is for preposition, JJ for adjunctive, and VB stands for verb phrase. The noun term written in bold and italic in the delineated patterns is the collocation of required entity of interest and that position is the most probable position having the named entity token. The process of pattern extraction process is shown in Fig 2.

Table 3. Part-of-Speech Context Patterns.

S no.	POS Pattern	Description	Example
1.	NN + IN + <i>NN</i>	Any noun phrase followed by a preposition followed by required entity of interest.	...characteristics of rice.. ...production of turnip... ...yield of paprika....
2.	<i>NN</i> + NN	Two consecutive nouns	...oats variety... ...almond cultivar...
3.	JJ + <i>NN</i>	Adjective followed by noun	...black pepper... ...grind spelt...
4.	VB + <i>NN</i>	Verb phrase followed by noun	...cultivated rice... ...produced Walnuts...
5.	<i>NN</i> + VB	Noun followed by verb	...bean planted... ...sage harvested...
6.	IN + <i>NN</i> + IN	Noun phrase sandwich between two preposition	...of garlic in... ...in hazelnut for... ...with pistachios from ...
7.	NN + IN + VB + <i>NN</i>	Required noun phrase followed after a triplet of noun, preposition and verb.	...growth of cultivated oats... ...sulphur at dried apricots... ...advantage in producing ginger...

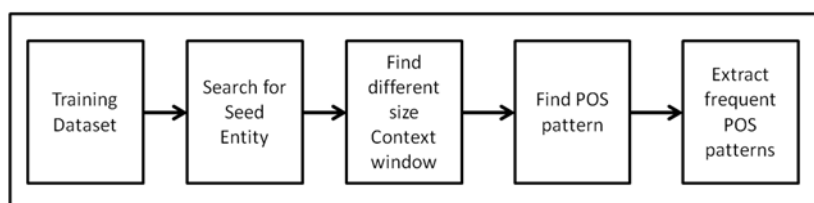


Fig. 2. Flow diagram of POS pattern Extraction.

Pattern Evaluation. In the pattern evaluation phase, the exactitude of the patterns which has been derived in the pattern extraction phase are to be enumerated. To check the soundness of these patterns a separate subset of data called the testing dataset is taken. Then the whole dataset is parsed for the POS structure. Once the POS sequence of words in the dataset is obtained, we will look for the sub sequence matching the POS patterns structure. After getting the POS pattern sequence, the word sequence corresponding to that POS sequence are chunked out. In this chunk of word, the token laying in the most probable position of NE in the POS pattern sequence is then extracted. The steps involved in the pattern evaluation phrase is illustrated in the Fig. 3.

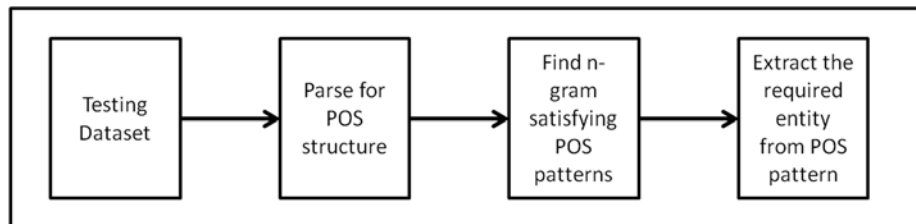


Fig. 3. Flow diagram of POS pattern Evaluation.

Pruning. It is supposed that the list of tokens obtained as an output in the previous section contain the named entity terms. However along with the required entity of interest there is a faction of irrelevant terms. In order to improve the accuracy of the system, these irrelevant terms must be filtered out from the required relevant terms. There are various ways to drive this for an instance semantic similarity between different set of words, using difference between the information content from any knowledge base [24] like Word net, concept net or any domain specific ontology, exercise different clustering algorithm like k means and many other. We have used semantic similarity measure to alienate the extraneous words from the list of words. There are also multifarious techniques available for scaling the semantic similarity between set of words. Bollegala [5] and Haiyan [10] employed Web search engine to measure the semantic similarity between words using web search engine. Pekar and Staab [22] and Pedersen *et al.* [21] uses the relatedness of concepts in the thesaurus and distributional feature vector to compute the semantic similarity. While Li *et al.* [15] employed multiple information sources for doing the same. In this work we have plied the UMBC semantic similarity model proposed by Han *et al.* [11]. The online demonstration of the model is available at <http://swoogle.umbc.edu/SimService/>. It is based on align-and-penalize algorithm. The algorithm works over the concept that if two sentences or short text sequences are semantically equivalent then align them and the words that are poorly aligned have to give penalty. The alignment quality serves as a similarity measure. The Semantic Textual Similarity (STS) is computed using:

$$STS = T - P' - P''$$

where T is the term alignments score, P' is the penalty for bad term alignments and P'' is the penalty for syntactic contradictions led by the alignments. However, P'' had not been fully implemented, it is shown just for completeness. Alignments score T is define by the equation:

$$T = \frac{\sum_{t \in S_1} sim'(t, g(t))}{2|S_1|} + \frac{\sum_{t \in S_2} sim'(t, g(t))}{2|S_2|},$$

where $g(t)$ is the aligning function:

$$g(t) = \arg \max sim'(t, t').$$

$sim'(t, t')$ is a wrapper function over the relation similarity model. The similarity between two words X and Y is computed by combining LSA similarity and Word Net relations:

$$sim_{\oplus}(x, y) = sim_{LSA}(x, y) + 0.5e^{-\alpha D(x,y)},$$

where $D(x, y)$ is the minimal path distance between x and y in the word net taxonomy.

For penalizing the bad terms, a model had been developed that determine whether two terms belong to disjoint sets:

$$A_i = \{ \langle t, g(t) \rangle | t \in S_i \wedge sim'(t, g(t)) < 0.05 \},$$

$$B_i = \{ \langle t, g(t) \rangle | t \in S_i \wedge t \text{ is an antonym of } g(t) \} \quad i \in \{1, 2\}.$$

P' is calculated using the formula:

$$P_i^A = \frac{\sum_{\langle t, g(t) \rangle \in A_i} (sim'(t, g(t)) + w_f(t) \cdot w_p(t))}{2|S_i|},$$

$$P_i^B = \frac{\sum_{\langle t, g(t) \rangle \in B_i} (sim'(t, g(t)) + 0.5)}{2|S_i|},$$

$$P' = P_1^A + P_1^B + P_2^A + P_2^B.$$

The semantic similarity value returned by the model among two terms ranges from 0 and 1. The similarity value increases from 0 to 1. In this work we have taken the threshold, τ as .4 for filtering out the irrelevant terms. Algorithm, shown in Table 4, presents the implementation detail of the proposed POS context pattern approach.

Table 4. Algorithm for POS Context Pattern.

Algorithm: Word Context Pattern.

Input: D_{TR} – Training Dataset, D_{TS} – Testing Dataset, S – Seed Word

Output: Agriculture Entities $E = \Phi$

POS Pattern Extraction

1. $E \leftarrow \Phi, T \leftarrow \Phi;$
 2. for each $s \in S$ do
 3. Find context window for W_i in D_{TR} where $i = \pm 5$
-

Algorithm: Word Context Pattern.

-
4. Extract co-occurred words, $CW_d(i) \leftarrow W_i$
 5. Get POS structure S_{POS} by parsing the co-occurred word
 $S_{POS} \leftarrow Parse(CW_d)$
 6. Part of speech patterns, $P_{POS} \leftarrow Frequent(S_{POS})$
- Pattern Evaluation**
7. For each sentence L_i in D_{TS} do
 8. Get POS structure S_{POS} by parsing each sentence L
 $S_{POS} \leftarrow Parse(L)$
 9. Search n-gram of part of speech p , where $p_{POS} \in S_{POS}$ and $p_{POS} := P_{POS}$
 10. Extract word token T from the probable position in p_{POS} .
- Pruning**
11. For each token $T_i \in T$ do
 12. Semantic Similarity, $Sim \leftarrow Semantic\ Similarity(C_i, T_i)$
 13. If $Sim \geq Threshold\ \Gamma$
 14. Agriculture Entity, $E := E + T$
-

The flow diagram of the proposed model for POS context pattern is shown in Fig. 4.

4 Experiment and Result

4.1 Dataset Preparation

As NER in agriculture domain is an unenlightened field, we have not found any standard or benchmark dataset. Hereof in order to perform the experiments we have to make our own dataset. To prepare the dataset a renowned agricultural resource called AGRIS is employed. AGRIS is a global public domain database with more than 7.5 million structured bibliographical records on agricultural science and technology (<https://en.wikipedia.org/wiki/AGRIS>). More specifically it is an agricultural search engine which contains the references to the agricultural research articles, data, statistics, and multimedia material. For preparing the dataset we have extracted out the abstracts of the research articles available in the AGRIS database. In this paper five different class entities have been sorted out, which is to be tagged by the proposed NER system. These predefined classes are: Name of cereals, fruits, nuts, spices, and vegetables. For each of these classes we have used 15 varied elements or entities for extracting the abstract. As a result of this we have a set of 75 entities. These seventy-five entities are passed to the AGRIS data base and for each entity top ten abstracts have been extracted. Hence in total, for performing the experiment we have a dataset of 750 documents or abstracts containing 171,735 words. Out of these 15 entities in each class we have used five elements for learning and ten elements for testing. Thus, in total we have used 25 entities or 250 documents for learning and 50 entities or 500 documents

for testing. Table 5 enlist the five classes and the entities contained in that classes in the testing dataset.

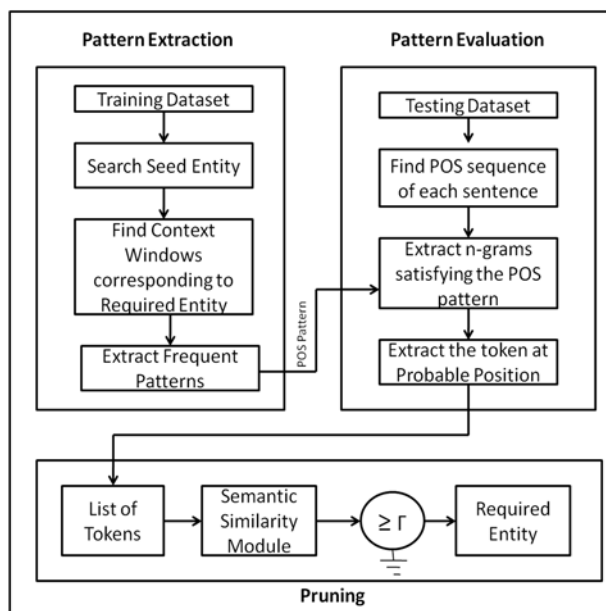


Fig. 4. Flow diagram of POS Context Pattern.

Table 5. Agriculture class Vs Entities per class.

		Class Name				
		Cereal	Fruit	Nut	Spice	Vegetable
Entity Name	Oats (101)	Acorn (72)	Almond (70)	Balm (30)	Carob (69)	
	Rice (323)	Apple (132)	Cashew (83)	Cinnamon (77)	Peanut (84)	
	Rye (129)	Apricot (57)	Chestnut (86)	Clove (66)	Cassava (25)	
	Wheat (325)	Blackberry (58)	Coconut (182)	Garlic (166)	Leeks (70)	
	Barley (193)	Durian (108)	Hazelnut (64)	Ginger (90)	Okras (18)	
	Bean (216)	Kiwi (48)	Pecan (26)	Mint (54)	Parsnips (18)	
	Corn (265)	Mango (116)	Pistachios (68)	Paprika (62)	Rhubarb (52)	
	Lentils (55)	Nectarine (47)	Walnut (72)	Pepper (98)	Tomato (67)	
	Spelt (87)	Pear (231)	Raisin (27)	Sage (64)	Turnip (28)	
		Plum (104)		Savory (38)		

4.2 Experiment and Result Analysis

In this paper the motto of the work is to extract the agricultural named entities from the dataset. Experiment 1 tries to find out these entities using the context pattern of words. For this at first the phrases or keywords are extracted which frequently co-occurred with the required named entities. Few of these extracted co-occurred word patterns have been enlisted below in Table 6.

To check the co-occurring accuracy of these word patterns with the agricultural entities, they are searched in the testing dataset and checked the token found in the probable position for named entity. However, after doing the experiment we found that the results obtained are not satisfactory. This is because of the reason that although the co-occurrence of these phrases is frequent with the agricultural entities but the probability of occurrence of these words patterns with the other entities is also very high.

Since the co-occurrence of word patterns are diverse in nature, no significant result has been obtained. Hence in order to devise more generalize co-occurring patterns we have move towards POS patterns.

Same as word patterns, for extracting the POS patterns also, n-gram of different size window is chunked out from the training data set. In this case rather than considering the word tokens the extracted n-gram phrases are parsed for the POS structure. For infesting the POS structure Stanford's parser has been used. As a result of this a set of finite POS patterns have been obtained, out of which some frequent POS patterns have been sorted out (Table 3).

To check the accuracy of the procured patterns, these POS structures have been searched over the testing dataset. The n-gram of word which satisfies the POS structure is then extracted out and looks for the token in the probable position. At last the list of tokens obtained through the probable position is check for the required entity of interest or not. The number of relevant entities extracted out through each pattern in each class is shown in Table 7.

Table 6. Co-occurred Word Patterns.

S. No.	n- gram	Co-occurring Word Patterns
1.	yield of rice	yield of ____
2.	content in apple	content in ____
3.	production of cashew	production of ____
4.	quality of tomatoes	quality of ____
5.	wheat production	____ production
6.	barley grain	____ grain
7.	wheat flour	____ flour
8.	Almond cultivars	____ cultivars
9.	bean varieties	____ varieties
10.	mint plant	____ plant
11.	corn yield	____ yield
12.	common wheat	common ____
13.	red rice	red ____
14.	wild lentil	wild ____
15.	raw barley	raw ____
16.	fresh paprika	fresh ____
17.	apples produced	____ produced
18.	ginger grown	____ grown
19.	cultivated rice	cultivated ____
20.	polished barley	polished ____

Table 7. Relevant vs. Irrelevant entities extracted through POS patterns without semantic similarity measure.

POS Pattern	Cereal		Fruit		Nut		Spice		Vegetable	
	Rv	IRv	Rv	IRv	Rv	IRv	Rv	IRv	Rv	IRv
NN + IN + NN	385	1055	180	690	173	634	143	576	158	584
NN + NN	866	2506	382	1864	347	161	208	1421	166	1271
JJ + NN	272	1048	135	874	109	691	128	782	36	635
VB + NN	154	501	70	403	71	391	15	75	40	310
NN + VB	225	1607	141	1266	116	986	140	1018	105	1105
IN + NN + IN	41	305	22	201	19	193	11	169	24	166
NN + IN + VB + NN	42	54	15	30	16	39	9	30	5	27

Here in Table 7, Rv referred to the relevant entities and IRv referred to the irrelevant entities extracted through patterns. Total number of entities extracted is the summation of all relevant and irrelevant entities extracted. The accuracy of the system can be measured using the formula:

$$\text{Precision} = \frac{\text{Total no. of relevant entities extracted}}{\text{Total no. of entities extracted}}$$

Numerated value of precision for each of the patterns is enlisted in Table 8.

Table 8. Precision value of POS patterns without similarity measure.

POS Pattern	Cereal	Fruit	Nut	Spice	Vegetable
NN + IN + NN	26.73	20.68	21.43	19.88	21.29
NN + NN	25.68	17.0	68.30	12.76	11.55
JJ + NN	20.60	13.37	13.62	14.06	5.36
VB + NN	23.51	14.79	15.36	16.66	11.42
NN + VB	12.28	10.02	10.52	12.08	8.67
IN + NN + IN	11.85	9.86	8.96	6.11	12.63
NN + IN + VB + NN	43.75	33.33	29.09	23.07	15.62

Fig. 5 represents the accuracy or precision curve for each of the proposed POS patterns shown in Table 8 for each of the varied class of agricultural elements without semantic similarity. It can be observed through Table 8 and the precision curve plotted in Fig. 5 that the precision value for the proposed POS patterns is not appreciable. The precision is lousy because of the occurrence of the irrelevant entities therewith the relevant entities. Hence in order to improve the accuracy of the system the irrelevant entities must be filtered out.

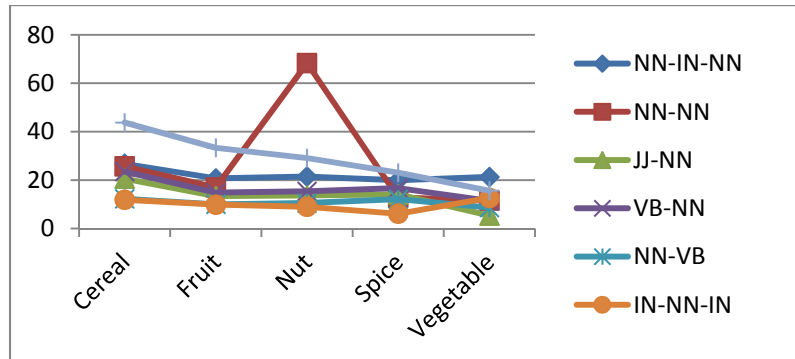


Fig. 5. Precision value of POS patterns (without semantic similarity measure).

To filter out the irrelevant entities from the list of extracted entities, semantic similarity measure has been used. We have employed the UMBC semantic similarity module to achieve this. The semantic similarity value of each term extracted out through patterns is measured with its class keyword that is cereal, fruit, nuts, spice, and vegetable. Terms for which the similarity measure is greater than or equal to .4 are kept as relevant named entities while the others are filtered out. The number of relevant and irrelevant entities obtained after the application of semantic similarity measure is enlisted in Table 9.

Table 9. Extracted Relevant vs. Irrelevant entities through POS patterns with semantic Similarity measure.

POS Pattern	Cereal		Fruit		Nut		Spice		Vegetable	
	Rv	Irv	Rv	Irv	Rv	Irv	Rv	Irv	Rv	Irv
NN + IN + NN	363	37	147	61	134	52	76	47	123	48
NN + NN	806	142	317	150	290	110	120	99	103	105
JJ + NN	161	33	115	40	45	20	81	19	20	21
VB + NN	147	34	58	21	64	18	32	20	32	20
NN + VB	220	64	105	98	92	60	68	51	97	44
IN + NN + IN	34	5	18	7	17	10	7	1	19	7
NN + IN + VB + NN	34	3	14	3	11	8	9	5	5	3

Table 10 represents the precision value of each of the proposed POS patterns derived from Table 9 for each of the varied class of agricultural elements with semantic similarity.

Table 10. Precision value of POS patterns with similarity measure.

POS Pattern	Cereal	Fruit	Nut	Spice	Vegetable
NN + IN + NN	90.75	70.67	72.04	61.78	71.92
NN + NN	85.02	67.88	72.5	54.79	49.51

JJ + NN	82.98	74.19	69.25	81.0	48.78
VB + NN	81.21	73.41	78.04	61.53	61.53
NN + VB	77.46	51.72	60.52	57.14	68.79
IN + NN + IN	87.17	72.0	62.96	87.5	73.07
NN + IN + VB + NN	91.89	82.35	57.89	64.28	62.5

The precision curve for the POS pattern with semantic similarity measure for each class of agriculture entity is illustrated in Fig 6.

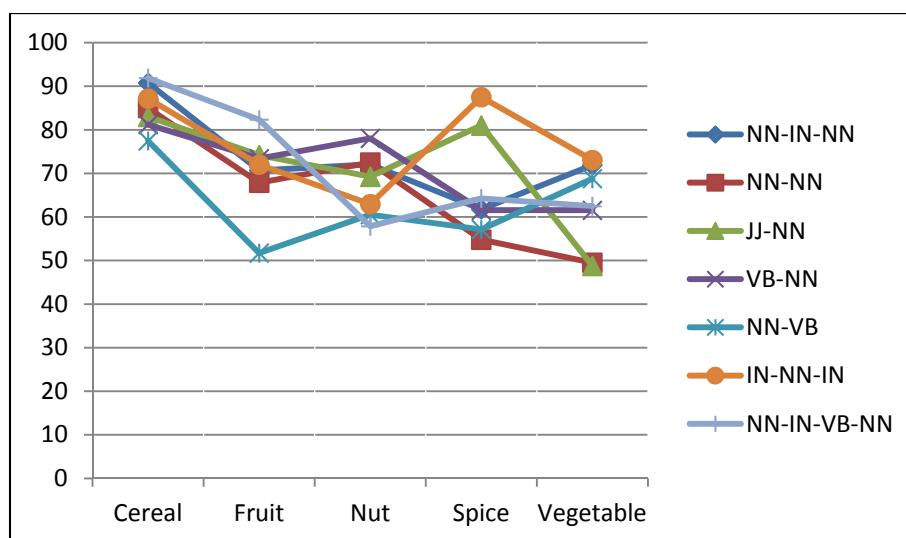


Fig. 6. Precision value of POS patterns (with semantic similarity measure).

It can look over through Table 8 and 10 and Fig. 5 and 6 that the application of semantic similarity has drastic effect on upgrading the performance of the system. The average precision of each pattern with and without similarity measure is framed in Table 11.

Table 11. Average precision for proposed POS patterns.

POS Pattern	Without Similarity Measure	With Similarity Measure
NN + IN + NN	22.00	73.43
NN + NN	27.05	65.94
JJ + NN	13.38	71.23
VB + NN	16.34	71.14
NN + VB	10.71	63.12
IN + NN + IN	9.88	76.54
NN + IN + VB + NN	28.97	71.78

The comparison of precision value of POS patterns for both that is without similarity measure and with similarity measure is shown in Fig. 7. As a result of application of

semantic similarity measure the average accuracy of the system increases from 18.33% to 70.45%. The recall value of the system can be scaled using the formula:

$$\text{Recall} = \frac{\text{Total no. of relevant entities extracted}}{\text{Total no. of relevant entities in the dataset}}$$

However, it cannot be calculated separately for each of the patterns. This is because of the overlapping nature of few of the POS patterns. Hence the recall value is accounted for the entire system rather than for varied separate patterns. In order to figure out this the entire data set is scanned for all the agriculture entity present in the data considering both hidden and known (fifty seed entities) agricultural entities. The recall is computed on the basis of out of all the relevant agricultural entities available in the data set, how many are extracted through patterns. It is found that there are 115 unique agricultural entities present in the data set and out of these 105 relevant agricultural named entities are extracted through patterns. Hence the recall value is obtained as 91.3%. The surpassing achievement is that the proposed patterns has extracted out fifty-three hidden agricultural named entities from the dataset in addition of fifty known seed entities.

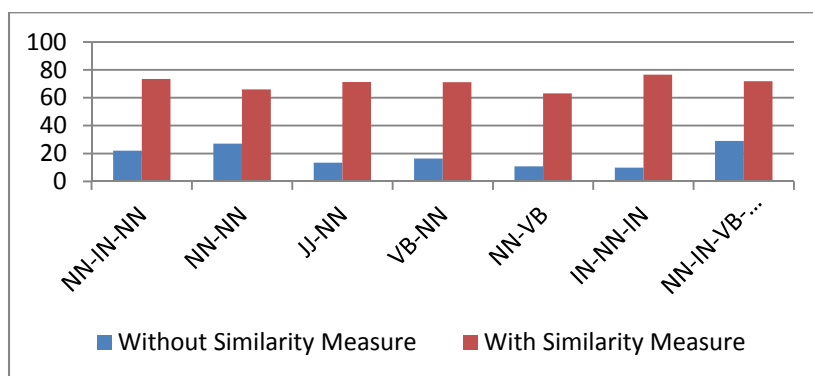


Fig. 7. Precision without similarity measure Vs Precision with similarity measure.

5 Conclusion

This paper focuses on developing an NER system for agriculture domain. For achieving this context pattern-based method has been used. The context has found using the co-occurrence pattern in different size window. Two type of context has been considered, word context pattern and POS context pattern. The output of the word context pattern is diverse in nature hence no significant result has obtained. Thus, the next experiment focuses on POS context pattern, which provide more generalize results. This approach receives good recall, but due to the impedance of bulk of irrelevant entities with the relevant ones, the accuracy of the system is very poor. Hence in order to filter out the irrelevant entities and improve the accuracy of the system semantic similarity measure is used. The experiment is deficient in a point that the semantic measure is relied over the word net database. Hence only those entities will be entertained for similarity measure which has enlisted in the word net database. Although the experimental results

are not very surprising, but being a preparatory work in the field of agricultural NER, this can be an impellent for future work in this blooming topic of NER in agriculture domain.

References

1. Al-Jumaily, H., Martínez, P., Martínez-Fernández, J.L., Van der Goot, E.: A real time Named Entity Recognition system for Arabic text mining. *Language Resources and Evaluation* 46(4), 543–563 (2012)
2. Biswas, P., Sharan, A., Kumar, A.: AGNER: Entity tagger in agriculture domain. In: *Proceeding of 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1134–1138 (2015)
3. Biswas, P., Sharan, A., Kumar, R.: Question Classification using syntactic and rule based approach. In: *Advances in Computing, Communications and Informatics (ICACCI, 2014) International Conference on. IEEE*, pp. 1033–1038 (2014)
4. Biswas P., Sharan A., Verma S.: Named entity recognition for agriculture domain using word net. *Int J Comput Math Sci.* 5(10), 29–36 (2016)
5. Bollegala D., Matsuo Y., Ishizuka M.: A web search engine-based approach to measure semantic similarity between words. *IEEE Transactions on Knowledge and Data Engineering* 23(7), 977–90 (2010)
6. Collier, N., Nobata, C., Tsujii, J.: Extracting the names of genes and gene products with a hidden Markov model. In: *Proceedings of COLING*, pp. 201–7 (2000)
7. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld D.S., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence* 165(1), 91–134 (2005)
8. Fleischman, M.: Automated subcategorization of named entities. In: *ACL (Companion Volume)*, pp. 25–30 (2001)
9. Grishman, R., Sundheim, B.: Message understanding conference-6: A brief history. In: *16th International Conference on Computational Linguistics*, Vol. 96, pp. 466–477 (1996)
10. Haiyan, C.: Measuring semantic similarity between words using web search engines. *Computer Science* 42(2), 261–267 (2015)
11. Han, L., Kashyap, A., Finin, T., Mayfield, J., Weese, J.: UMBC_EBIQUITY-CORE: Semantic textual similarity systems. In: *2nd Joint Conference on Lexical and Computational Semantics (*SEM)*, Vol. 1, pp. 44–52 (2013)
12. Kazama, J., Makino, T., Ohta, Y., Tsujii, J.: Tuning support vector machines for biomedical named entity recognition. In: *Proceedings of the workshop on natural language processing in the biomedical domain*, pp. 1–8 (2002)
13. Lamurias, A., Ferreira, J., Couto, F.M.: Chemical named entity recognition: Improving recall using a comprehensive list of lexical features. In: *8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014)*, pp. 253–260. Springer International Publishing (2014)
14. Lee, S., Lee, G.: Heuristic methods for reducing errors of geographic named entities learned by bootstrapping. In: *International Joint Conference on Natural Language Processing*, pp. 658–669. Springer, Berlin, Heidelberg (2005)
15. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering* 15(4), 871–82 (2003)
16. Lin, Y.F., Tsai, T.H., Chou, W.C., Wu, K.P., Sung, T.Y., Hsu, W.L.: A maximum entropy approach to biomedical named entity recognition. In: *Proceedings of the 4th International Conference on Data Mining in Bioinformatics*. pp. 56-61. Springer-Verlag (2004)
17. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Human

- Language Technologies, Vol. 1, pp. 359–367. Association for Computational Linguistics (2011)
18. Malarkodi, C.S., Lex, E., Devi, S.L.: Named Entity Recognition for the Agricultural Domain. *Research in Computing Science* 117, 121–132 (2016)
 19. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*-Vol. 4, pp. 188–191. Association for Computational Linguistics (2003).
 20. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30(1), pp. 3–26 (2007)
 21. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet: Similarity: measuring the relatedness of concepts. In: *Demonstration papers at hlt-naacl 2004*, pp. 38–41. Association for Computational Linguistics (2004)
 22. Pekar, V., Steffen, S.: Word classification based on combined measures of distributional and semantic similarity. In: *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, Vol. 2*, pp. 147–150. Association for Computational Linguistics (2003)
 23. Rau, L.F.: Extracting company names from text. In: *7th IEEE Conference on Artificial Intelligence Application, Vol. 1*, pp. 29–32 (1991)
 24. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007* (1995)
 25. Ritter, A., Clark, S., Etzioni, O.: Named entity recognition in tweets: an experimental study. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1524-1534. Association for Computational Linguistics (2011)
 26. Rizzolo, N., Roth, D.: Learning Based Java for Rapid Development of NLP Systems. In: *LREC Vol. 5*, pp. 313–323 (2010)
 27. Saha, S.K., Sarkar, S., Mitra, P.: Feature selection techniques for maximum entropy based biomedical named entity recognition. *Journal of biomedical informatics* 42(5), 905–911 (2009)
 28. Seker, G.A., Eryigit, G.: Initial Explorations on using CRFs for Turkish Named Entity Recognition. In: *COLING*, pp. 2459–2474 (2012)
 29. Sun, C., Guan, Y., Wang, X., Lin, L.: Rich features based Conditional Random Fields for biological named entities recognition. *Computers in Biology and Medicine* 37(9), 1327–1333 (2007)
 30. Takeuchi, K., Collier, N.: Use of support vector machines in extended named entity recognition. In: *Proceedings of the sixth conference on natural language learning (CONLL)*, pp. 119–25 (2002)
 31. Tanabe, L., Wilbur, W.J.: Tagging gene and protein names in biomedical text. *Bioinformatics* 18(8), 1124–1132 (2002)
 32. Tohidi, H., Ibrahim, H., Murad, M.A.A.: Improving named entity recognition accuracy for gene and protein in biomedical text literature. *International journal of data mining and bioinformatics* 10(3), 239–268 (2014)
 33. Tsai, R.T.H., Wu, S.H., Chou, W.C., Lin, Y.C., He, D., Hsiang, J., Sung, T.Y., Hsu, W.L.: Various criteria in the evaluation of biomedical named entity recognition. *BMC bioinformatics* 7(1), p. 92 (2006)
 34. Zhang, J., Shen, D., Zhou, G., Su, J., Tan, C.L.: Enhancing HMM-based biomedical named entity recognition by studying special phenomena. *Journal of biomedical informatics* 37(6), 411–422 (2004)
 35. Zhou, G.D., Su, J.: Named Entity Recognition using an HMM-based Chunk Tagger. In: *Proceedings of the 40th annual meeting of the association for computational linguistics (ACL)*, pp. 473–480 (2002)

Electronic edition
Available online: <http://www.rcs.cic.ipn.mx>

