

Advances in Computing Science

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Alexander Gelbukh (Mexico)
Ioannis Kakadiaris (USA)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

María Fernanda Ríos Zacarias

Formatting:

Juan Carlos Chimal Eguía
Erandi Castillo Montiel
Iliac Huerta Trujillo
Julio Cesar Rangel Reyes
Sarahi Sanchez-Piña
Angel Refugio Mandujano Garcia

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 107**, noviembre 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsible: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 107**, November 2015. Printing 500. The authors are responsible for the contents of their articles. 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. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 107

Advances in Computing Science

**Juan Carlos Chimal Eguía,
Erandi Castillo Montiel,
Iliac Huerta Trujillo,
Julio Cesar Rangel Reyes (eds.)**



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



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2015

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 and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Editorial

The purpose of this volume is to present the most recent advances in selected areas of Computer Science. The works included in this volume were carefully selected by the editors on the basis of the blind reviewing process being the main criteria for selection the originality and technical quality. This issue of the journal *Research in Computing Science* will be useful for researches and students working in various areas of Computer Science, as well as for all readers interested in enrichment of their knowledge in this field.

All received papers that were submitted for evaluation for the special issue were reviewed by 2 independent members of the editorial board of the volume or additional reviewers. In general, the acceptance rate was 48%. This volume contains revised and corrected versions of 15 accepted papers.

We would like express our gratitude to all people who help to elaborate this volume. First, the authors of the papers for the technical excellence of their works that guarantees the quality of this publication. We also want to thank the members of the editorial board for their hard work in evaluating and selecting the best papers out of many submissions that we received. We express sincerely our gratitude to the Sociedad Mexicana de Inteligencia Artificial (SMIA) for its collaboration in preparation of this publication. Also we want to give special recognition to the Centro de Investigación en Computacion of the Instituto Politécnico Nacional (CIC-IPN) for their support in publication of this volume. The submission, reviewing, and selection process was supported for free by the EasyChair system, www.EasyChair.org.

Juan Carlos Chimal Eguía
Erandi Castillo Montiel
Iliac Huerta Trujillo
Julio Cesar Rangel Reyes

November 2015

Table of Contents

	Page
Modelado y propagación de valores de sentimiento en relaciones de usuario.....	9
<i>Ramón Rivera Camacho, Ricardo Barrón Fernández, Adolfo Guzmán Arenas</i>	
Optimal Synthesis and 3D Modeling of a Lifting Mechanism for a Platform with Variable Slope	19
<i>Eduardo Vázquez-Santacruz, Cuauhtémoc Morales-Cruz, Rogelio de Jesus Portillo-Velez, Mariano Gamboa-Zúñiga</i>	
Programming by Demonstration in DLProlog.....	31
<i>José Oscar Olmedo-Aguirre, Marisol Vázquez-Tzompantzi, Giner Alor-Hernández</i>	
Propuesta de artefactos basados en una notación con grafos y conjuntos para el modelado conceptual de aplicaciones Web	41
<i>Carlos Daniel Lima Gámez, Silvestre Gustavo Peláez Camarena, Ulises Juárez Martínez, María Antonieta Abud Figueroa, Celia Romero Torres</i>	
ReAHK: Sistema auxiliar para la rehabilitación articular del hombro mediante Kinect	51
<i>Alan J. Crisantos Bazaldúa, Ángel E. Hernández Beltrán, Eusebio Ricárdez Vázquez</i>	
Reconocimiento de patrones numéricos para vuelo controlado de un AR drone utilizando redes neuronales artificiales.....	61
<i>Juan Carlos Rodríguez-Sánchez, Victor Manuel Landassuri-Moreno, José Martín Flores Albino</i>	
Red de sensores inalámbrica para la detección de incendios forestales.....	73
<i>Jorge Antonio Atempa Camacho, Arnoldo Díaz-Ramírez, Rafael Iván Ayala Figueroa, Verónica Quintero Rosas</i>	
Simulación 3D basada en agentes de robo y asalto a transeúntes	87
<i>Fabián Beltrán Osnaya, Héctor Rafael Orozco Aguirre, Victor Manuel Landassuri Moreno</i>	
Sistema de apoyo lingüístico en español para personas sordas	99
<i>Obdulia Pichardo-Lagunas, Beatriz Torres-Alatriste, Bella-Citlali Martínez-Seis, Victor-Darío Cuervo-Pinto, Miguel-Alejandro Martínez-Rosales</i>	

Sistema de monitoreo remoto de sensores de temperatura y corriente usando una red hibrida bus CAN-Zigbee.....	111
<i>Héctor Anzures Martínez, Ángel Bulmaro Sánchez Rubio, Josefina Castañeda Camacho, Gerardo Mino Aguilar, José Fermi Guerrero Castellanos, Liliana Cortez</i>	
Sistema embebido para la detección de luz láser empleando el soft-core Nios II ...	121
<i>Julio C. Sosa, Iván Dominguez-Lopez, Adrián L. García-García, J. D. Oscar Barceinas-Sánchez, Anuar Jassen</i>	
Solución al problema de ordenación usando PipeLine y Binary Tree como composiciones paralelas de alto nivel.....	133
<i>Mario Rossainz-López, Ivo H. Pineda-Torres, Patricia Domínguez Mirón</i>	
Un sistema experto difuso en la Web para diagnóstico de diabetes.....	145
<i>Viridiana Cruz-Gutiérrez, Abraham Sánchez-López</i>	
Using the M-Learning to Support Learning of the Concept of the Derivative	157
<i>Daniel A. Sarmiento, Chadwick Carreto A., Elena F. Ruiz</i>	
3-approximation Algorithm for the Travelling Repairman Problem with Unit Time-windows	167
<i>Luis Eduardo Urbán Rivero, Cynthia A. Rodríguez Villalobos, Rafael López Bracho, Francisco Javier Zaragoza Martínez</i>	

Modelado y propagación de valores de sentimiento en relaciones de usuario

Ramón Rivera Camacho, Ricardo Barrón Fernández, Adolfo Guzmán Arenas

Instituto Politécnico Nacional, Centro de Investigación en Computación,
México

Resumen. Se propone un método de modelado de relaciones personales basado en dos aspectos principales, la relevancia y componente emocional subjetiva del usuario y la categorización objetiva de las relaciones. Se presenta dicho modelo como un grafo conexo con la suma de todos los componentes como nodo principal rodeado de relaciones paradigma y relaciones descubiertas conectadas a través de niveles de relevancia. En segunda instancia se describe la manera de propagar el componente sentimental a través del modelo construido por medio del método de retropropagación (*back-propagation*). Finalmente se presenta las propiedades de dicho modelo, resultados obtenidos y posibles usos del mismo.

Palabras clave: Análisis de sentimientos, similitud semántica, aprendizaje automático.

Modelling and Propagation of Sentiment Values in Relations between Users

Abstract. A method of modeling personal relationships is proposed grounded on two main aspects, the subjective relevance and emotional component of the user, and the objective categorization of their relationships. This model is built as a connected graph with its principal node depicting the complete sum of all of the emotional components surrounded by paradigm and discovered relationships joined to this according to their level of relevance. As a compliment, a way to spread the sentimental component through the model by back-propagation method is described. Finally, some properties of the model are shown, as well as obtained results and possible usages of this.

Keywords: Sentiment analysis, semantic similarity, machine learning.

1. Introducción

A medida que la cantidad de información textual incrementa dentro de la red, el sector dedicado a la minería de opiniones, también conocido como análisis de sentimiento, obtiene mayor importancia, ya que en este recaen las tareas de obtención de características y tópicos, así como la clasificación de las preferencias de usuario para éstos, a menudo obtenidos desde redes sociales, blogs de opiniones, entre otros; sin

embargo, poco desarrollo se ha realizado para analizar sentimientos que afectan al usuario en sí, así como el efecto en sus relaciones.

El análisis de sentimientos principalmente está enfocado a reportar la actitud de un sujeto con respecto a un objeto determinado, esta tarea puede realizarse, en general, de dos maneras, supervisada y no supervisada, siendo la primera la más precisa debido a que los clasificadores son entrenados con un conjunto de datos representativos, el corpus. Pueden destacarse dentro de este ramo las técnicas de aprendizaje automático (*Machine learning*), como el probabilístico Naïve Bayes que es un método relativamente sencillo pero que se caracteriza por realizar una buena clasificación de sentimientos, sobre todo cuando las características son altamente dependientes [2-3], y el basado en hiperplanos de categorización, Máquinas de Soporte Vectorial (*Support Vector Machines, SVM* por sus siglas del inglés), siendo este último el que mejor desempeño reporta [6].

En contraparte, los métodos no supervisados exhiben un menor desempeño pero minimizan el trabajo de clasificación previa y de dependencia de dominio, algunos de ellos trabajando por medio de palabras semilla (*Seed words*) y calculando la orientación semántica de las frases [6-8].

Conforme al aumento en la cantidad de datos es necesario que el análisis tenga una reducción en la dimensión de búsqueda, es decir, la extracción de tópicos. Mei, et al., asume la existencia de una mezcla de ellos en el documento y que por lo tanto pueden ser reducidos utilizando una distribución de probabilidad [5], mientras que Blei, et al. [4], se basa en el principio de intercambiabilidad de De Finetti para construir el modelo probabilístico. En conclusión, ambos basan su funcionamiento en la construcción de modelos probabilísticos generativos cuya tarea es encontrar los tópicos latentes ocultos dentro del documento.

Siguiendo la misma línea, además de utilizar las palabras semilla, pero enfocado a un escenario más general Lin et al. [1] proponen un conjunto de palabras con contenido sentimental conocido, la palabras paradigma, para clasificación de sentimientos no supervisada a nivel de documento, la cual realiza la extracción de tópicos por medio de clasificadores bayesianos, sin embargo, al igual que Blei [4] y Mei [5], el modelo representa los resultados como bolsas de palabras (*bag-of-words*), que no presentan relación entre ellos.

Sin embargo, la falta de relación entre resultados presenta inconsistencia para la interpretación de éstos, necesitando otro nivel de clasificación extra si el problema requiere de éste tipo de información. Blei, et al. [9], retoma la idea del proceso estocástico de restaurante chino (*Chinese restaurant process*), para inferir la jerarquía de los datos a analizar. Por otro lado, para lidiar con este problema, Kim, et al. [10], basándose en la idea que un documento presenta aspectos que pueden ser organizados de manera natural, construyen un árbol de aspectos-sentimientos que describen dicha relación.

En el presente artículo se considera la etapa final del problema de describir el cambio de humor del usuario por medio de valores de sentimiento dentro del conjunto de relaciones de usuario.

Se retoman las ideas previamente tratadas para realizar el modelado del estado de humor del usuario, asumiendo por definición las relaciones de parentesco del mismo, las relaciones paradigma, y considerando la afectación sentimental por medio de la propagación de componentes sentimentales. La descripción se centra en la construcción

de un modelo jerárquico por medio de distancias semánticas y la probabilidad de pertenecer a cierto conjunto superior, además, se propone un método de propagación del componente sentimental hasta alcanzar el nodo raíz por medio de técnicas bayesianas.

2. Análisis

Se aborda, como primer acercamiento, el modelado del cambio emocional debido a factores externos, con la construcción de un grafo que satisfaga la siguiente idea sencilla: “La descripción sentimental del usuario parte de la afectación de tópicos externos y mientras éste se refiera más a un tópico en específico, más relevancia sentimental tendrá y más cambios realizará en el usuario”.

Para la implementación de esta idea se construirá de manera no supervisada un modelo jerárquico que contenga las relaciones de parentesco del usuario, a partir de un conjunto de relaciones base que se complementarán con base en la similitud de nuevos términos, posteriormente, el valor de sentimiento que contenga se propagará hasta llegar al nodo principal tomando en cuenta la relevancia del nodo padre, obtenida por medio de la cantidad de veces que el tema haya sido tratado previamente. Este proceso es dividido en 2 etapas, la complementación y construcción del grafo que describe las relaciones y la propagación de las componentes de sentimiento.

Para crear el modelo de relaciones de usuario (URM), se utiliza un grafo no dirigido que contiene como nodos al conjunto de valores de relaciones conocidos, las relaciones paradigma (PRs), dadas por definición y que además presentan la propiedad es-parte-de (*part-of*) o pertenencia (\in) con respecto al conjunto que define su nodo padre [11], por ejemplo:

Padre(s) es-parte-de Familia(s) \rightarrow Padre(s) \in Familia(s)
 Novio(s) es-parte-de Conocido(s) \rightarrow Novio(s) \in Conocidos(s)

Donde, los nodos Familia y Conocidos mantiene la misma relación con un padre y éstos a su vez repiten el proceso con nodos de orden superior.

En primera instancia se debe partir de un nodo principal que define el total de las contribuciones de componentes sentimentales y el punto de partida para realizar la búsqueda dentro de las relaciones paradigma, el nodo “TODO” (“ALL”).

Por otro lado, aquellos términos que presenten valores de sentimiento y relevancia significantes, pero que no estén presentes dentro del URM serán clasificados tomando en cuenta distancias semánticas que evalúan el parentesco de dos conceptos evaluando los enlaces semánticos es-un (*is-a*) y parte-de (*part-of*) utilizando parámetros como la distancia más corta entre conceptos o la distancia a un concepto superior común (*subsumer*), en lo posterior tratado como *mscs* (Del inglés, *the most specific common subsumer*).

Se propone utilizar la medida de similitud semántica de Wu-Palmer [12], descrita en (1).

$$sim_{wp}(c_i, c_l) = \frac{2 * \min_{p \in pths(mscs(c_i, c_j), rt)} len_e(p)}{\min_{p \in pths(c_i, c_j)} len_e(p) + 2 * \min_{p \in pths(mscs(c_i, c_j), rt)} len_e(p)} \quad (1)$$

donde:

rt: Raíz ontológica

pths(x, y): Conjunto de caminos entre los conceptos x e y.

len_c(x): Cantidad de aristas hacia el camino x.

A diferencia de otras medidas de similitud semántica, Wu-Palmer toma en cuenta la distancia al mscs y no presenta necesidad de un corpus de entrenamiento, como otras presentadas en la literatura [12-15]. Realizando una búsqueda por anchura (BFS) y comparando esta similitud con los de un umbral definido por el usuario, para decidir la pertenencia a uno o varios nodos padre, generando las relaciones orden N (Nth-order discovered relationships, NDORs). Como se muestra en la figura 1:

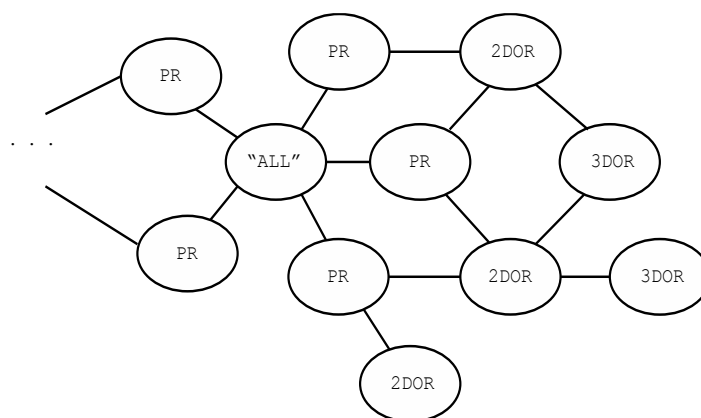


Fig. 1. Posible distribución del URM, se muestra el nodo principal “ALL”, rodeado de relaciones paradigma (PR) y relaciones descubiertas de orden N (NDOR).

Para realizar el cálculo de la componente emocional adjunta a la relación, utilizamos el método de propagación hacia atrás (*backpropagation*) del valor de sentimiento V_s considerando la relación jerárquica y semántica del portador con los nodos relacionados de orden superior, como se describe:

1. Sea un C_j la probabilidad de propagar un valor de sentimiento N_{act} a través de un conjunto finito de relaciones jerárquicas $N_r = \{N_{r1}, N_{r2}, \dots, N_{rn}\}$ de orden superior alcanzables, dada por:

$$P(C_j) = P(N_{act}|N_r) \quad (2)$$

$$P(N_p|N_{rel}) = \frac{P(N_{rel}|N_p)P(N_p)}{P(N_{rel})} \quad (3)$$

$$P(N_{rel}) = P(N_{os}|N_r) = \frac{P(N_r|N_{os})P(N_{os})}{N_r} \quad (4)$$

donde:

N_{os} = Nodos de orden superior, compuestos por los nodos directos, N_d , y los nodos alcanzables N_r .

N_{act} = Nodos actuales, compuestos por el nodo de propagación N_p y los nodos relacionados, N_{rel} .

2. Sea C_s la razón contribución semántica del portador del componente sentimental, N_{act} , con respecto al conjunto finito de relaciones directas de orden superior $N_p = \{N_{p1}, N_{p2}, \dots, N_{pn}\}$, por lo tanto:

$$C_s = C_{s1} + C_{s2} + \dots + C_{sn-1} \quad (5)$$

con:

$$C_{si} = \frac{N_{act|N_{pi}}}{N_{act|N_{p1}} + N_{act|N_{p2}} + \dots + N_{act|N_{pn}}} \quad (6)$$

donde:

N_{act} = Nodo portador de la componente sentimental.

$N_{p1}, N_{p2}, \dots, N_{pn}$ = Nodos relacionados directamente con N_{act} .

$N_{act|N_p}$ = Razón de valor semántico entre N_{act} y N_p .

Finalmente, la propagación se dará sucesivamente por medio de la función F_p , previamente descrita, dada por:

$$F_p \rightarrow V_S * C_s * C_j \quad (7)$$

Por lo tanto, el proceso se resume como un método de “complementación-propagación” de N número de objetos sentimiento-relación; asumiendo un conjunto de relaciones candidato (PRs), con una relevancia suficiente y conteniendo tanto el valor de componente sentimental a propagar como la relación adjunta, se propone el siguiente algoritmo de construcción del URM.

```

URM          (Modelo de relaciones de usuario)
PRs          (Conjunto de candidatos a agregarse al MRU)
Sim-umbral  (Umbral de similitud)
robjs       (Objetos de relación)
parents      (Nodos padre)
    
```

```

sub add_to_URM(robjs)
parents = []

for each robj in robjs
    if robj in URM
        URM[robj]->count = URM[robj]->count + 1
        backpropagation (URM[robj]);
    else
        if robj in PRs
            PRs[robj]->count = PRs[robj]->count + 1

            if PRs[robj] >= sim-umbral;
                parents = calculate_distances()
                attach_parents(PRs[robj], parents)
                backpropagation(PRs[robj])
    
```

```

        end if
    else
        add(robj, PRs)
        PRs[robj]->count = 1
    end if
end if
end for

```

donde las rutinas *attach_parents(PRs[obj], parents)* y *backpropagation()*, simplemente realizan el enlace con los nodos de orden superior y la propagación del componente de sentimiento como fue descrito respectivamente y *add(robj, PRs)* solamente agrega el objeto de entrada a las relaciones candidato.

3. Resultados

La presente sección se enfocará a analizar los casos que se presentan en la propagación del componente sentimental, cabe mencionar que no está dentro del alcance la obtención de los objetos de sentimiento, los cuales están definidos como las tuplas (n, p, v) , que representan (nombre, POS, valor).

Para tener control sobre la disipación de la componente en cada escenario, se propone un URM compuesto solamente por relaciones paradigma sin relevancia previa.

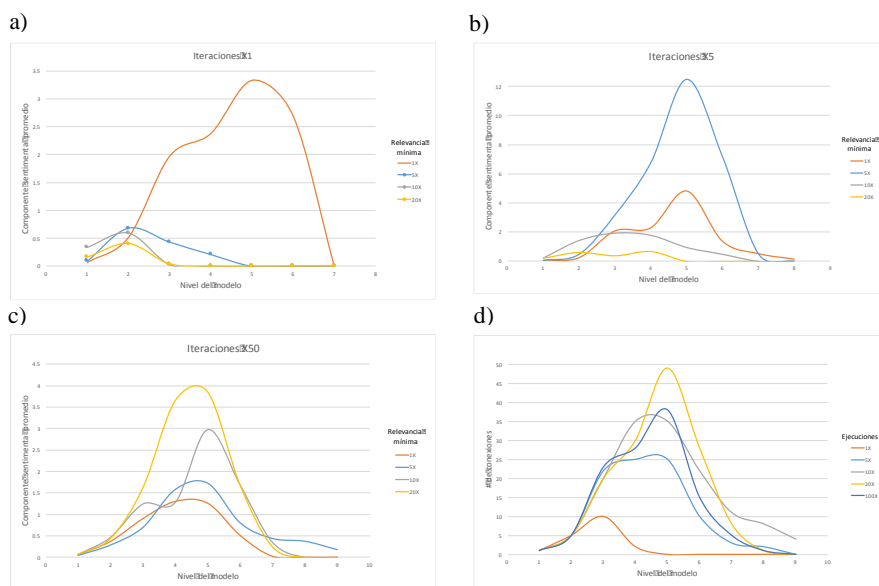


Fig. 2. Componente sentimental promedio a través de los niveles del URM para (a) $i=\{1iX\}$ y $r=\{1rX, 5rX, 10rX, 20rX\}$, (b) $i=\{5iX\}$ y $r=\{1rX, 5rX, 10rX, 20rX\}$, (c) $i=\{50iX\}$ y $r=\{1rX, 5rX, 10rX, 20rX\}$. (d) Número de relaciones por nivel para una relevancia fija $r = \{5rX\}$ e $i = \{1iX, 5iX, 10iX, 20iX, 100iX\}$.

Además se provee de un pool de objetos de sentimiento de prueba vagamente clasificados con pertenencia a las principales relaciones paradigma adjuntas al nodo principal, éstos objetos son los que se espera generen una componente sentimental con su correspondiente propagación dentro del modelo, aunados a ellos se suplen un conjunto de objetos aleatorios con componente sentimental pero que no son compatibles con el URM.

Se estipulan valores base para cantidad de iteraciones (iX) y valores de relevancia (rX) y se desarrollan pruebas con valores múltiplo de éstos para los siguientes escenarios, con las especificaciones antes descritas:

1. Tuplas (n, p, v) , donde: $(n \in \text{URM}, v \in [-3, 3])$, ejecuciones de prueba $i = \{1iX, 5iX, 50iX\}$, graficados para relevancia $r = \{1rX, 5rX, 10rX, 20rX\}$
2. Tuplas (N, p, v) , donde: $N = \{n, \bar{n}\} | \{n \in \text{URM}, \bar{n} \notin \text{URM}, |n| = |\bar{n}|\}$, $v \in [-2, 2]$, ejecuciones de prueba $i = \{3iX, 5iX, 10iX\}$ con relevancia fija de $r = 10rX$.

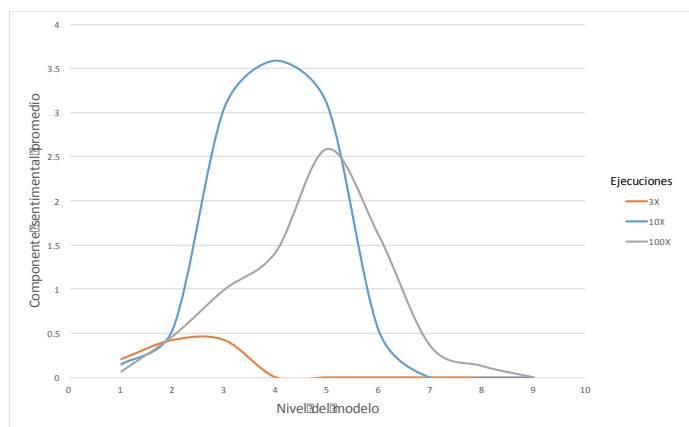


Fig. 3. Componente sentimental promedio para $N = \{n, \bar{n}\} | \{n \in \text{URM}, \bar{n} \notin \text{URM}, |n| = |\bar{n}|\}$ con $i = \{3iX, 5iX, 10iX\}$ y $r = 10rX$.

De los resultados se señalan los siguientes puntos:

1. La distribución de los valores de sentimiento a través de los niveles del modelo es similar independientemente de la cantidad de ejecuciones.
2. La cantidad de niveles aumenta con respecto se ejecute el modelo, de igual manera, a medida que se aumenten las ejecuciones la relación de candidatos debe tender a cero.
3. En niveles centrales del grafo se encontrarán el mayor número de conexiones y, por ende, también se presentará mayor relevancia en estos niveles
4. El parámetro de relevancia de un objeto de sentimiento impacta directamente en la construcción del modelo de relaciones, necesitando mayor cantidad de ejecuciones.

En los resultados presentados las relaciones son seleccionadas de manera aleatoria, por lo cual el modelo debería complementarse en anchura, sin embargo, en escenarios de usuario en los cuales se tratan con más cantidad menos tópicos, los niveles no

crecerán de manera tan uniforme, por lo tanto se podría ver con más claridad una tendencia en cuanto a tópicos con mayor relevancia subjetiva para el usuario.

Además, ya que se tiene un grafo conexo cuyo nodo principal es la representación del total de la suma de relaciones de usuario, dada una relación aleatoria es posible obtener la ruta hacia este nodo principal respecto a la relevancia de las relaciones, es decir, obtener el camino más probable de afectación sentimental de la relación.

4. Conclusiones y trabajo futuro

En este artículo, se define un primer enfoque al problema de modelar el cambio de humor debido a factores externos de un usuario. Se propone la construcción de un modelo jerárquico cuyos nodos exhiban la propiedad parte-de (part-of) con el fin de mantener una conexión entre las relaciones descubiertas y las dadas por definición, es decir, relaciones de orden N (N-ORs) y relaciones paradigma (PR), una vez encontrado el nodo, se presenta el método de propagación hacia atrás (*backpropagation*) con el fin de modificar los valores sentimentales de nodos de orden superior, en base a la relevancia del tópico que estos contienen.

Se mostraron las contribuciones en cuanto a relevancia y componente sentimental a través del modelo respecto a variaciones de umbral de relevancia y ejecuciones del algoritmo, asimismo, se presenta una alternativa para perseguir las relaciones con más afectación sentimental en base a la relevancia.

Las ampliaciones al presente trabajo se centran en las siguientes tareas, (1) saciar las premisas asumidas de la existencia de los objetos tópico-sentimiento, (2) reportar resultados con distintas medidas de similitud semántica, (3) generar de manera automática los categorizadores, (4) optimizar la propagación de los componentes sentimentales.

Referencias

1. Lin, C., He, Y.: Joint Sentiment/Topic Model for Sentiment Analysis (2009)
2. Domingos, P., Pazzani, M.J.: On the optimality of the Simple Bayesian classifier under Zero-One loss, *Machine Learning*, 29 (2–3) (1997)
3. Lewis D.D.: Naive (Bayes) at forty: The independence assumption in information retrieval. In: Proc. of the European Conference on Machine Learning (ECML) (1998)
4. Blei, D.M. Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation (2003)
5. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.X.: Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs (2007)
6. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of EMNLP (2002)
7. Turney, P.D., Littman, M.L.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus (2002)
8. Zagibalov, T., Carroll, J.: Automatic seed word selection for unsupervised sentiment classification of Chinese text. In: Proceedings of the 22nd International Conference on Computational Linguistics, Vol. 1 (2008)

9. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The Nested Chinese Restaurant Process and Bayesian Nonparametric Inference of Topic Hierarchies (2010)
10. Kim, S., Zhang, J., Chen, Z., Oh, A., Liu, S.: A Hierarchical Aspect-Sentiment Model for Online Reviews (2013)
11. Rada M., Radev, D.: Graph-Based natural language processing and information retrieval (2011)
12. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics (1994)
13. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics* 19 (1989)
14. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Vol. 1 (1995)
15. Blanchard, E., Harzallah, M., Briand, H., Kuntz, P.: A typology of ontology-based semantic measures (2005)

Optimal Synthesis and 3D Modeling of a Lifting Mechanism for a Platform with Variable Slope

Eduardo Vázquez-Santacruz, Cuauhtémoc Morales-Cruz,
Rogelio de Jesus Portillo-Velez, Mariano Gamboa-Zúñiga

Cinvestav-IPN, CGSTIC, Mexico, D.F., Mexico

{efvazquez, cmorales, mgamboaz}@cinvestav.mx

Abstract. Today, one of the most important research fields is concerned to the Engineering Design Process. It is focused on designing better electro-mechanical systems with a low impact on the environment and a most efficient use of the energy. Currently, the design process can take advantage of many computational-aided tools, which play a crucial role in the modern methods of optimization and reduce the cost of prototyping. On the other hand, a lifting mechanism for a platform that changes its slope in a specified range and moves along this restriction with a distributed load has many applications in different fields. In this work, the synthesis of a lifting mechanism is obtained by proposing and solving a design problem as a nonlinear optimization problem.

Keywords: Lifting mechanism, optimal parameterization, kinematic model.

1 Introduction

New fields and approaches have emerged to fulfill the design requirements of many disciplines. Medical requirements have been always at the top of demanding fields to satisfy hospitals requirements and more important to improve the patient comfort and quality of the devices that help with his recovery. New mechanical devices must be designed to fulfill successful health-care services. One of the devices that most impact has been the medical hospital bed [10, 9]. Several mechanical designs of hospital beds have been proposed to alleviate the very intensive labor and the lack of qualified personnel (nurses and stretcher-bearers), mainly in developing countries. In Ching-Hua [3], a hospital bed with auxiliary functions of lateral positioning and transferring patients is described. Three mechanisms are described which assist in complicated maneuvers of moving patients from hospital beds to the stretcher. Andhare [1] makes a design that attempt is to reduce the amount of assistance required in managing these patients. Special focus is made on the mechanism synthesis stage. Kittipichai [5], proposed an optimization procedure for the structure design of a hospital bed using genetic algorithms. The novelty of the bed structure is that it can support the left and or right leg for patient's leg splint. Some goals of the design are to reduce the mass of the bed structure. In the work of Shih-Wei [7] the

Mechanism Design and Mechatronic Control of a Multifunctional Test Bed for Bedridden Healthcare is presented. The design considers two beds, one main bed, and one nursing bed with transferring capabilities. A remarkable feature is that the designed bed is built by mechatronic engineers and qualified healthcare personnel simultaneously. Kap-Ho [4] presents the development of an intelligent bed robot system, in particular, a bed equipped with two robot arms and an array of pressure sensors. These sensors are attached to the mattress capable of estimate the pose of the patient. Mohammed [6] present a new design of a Multi-Functional Portable Patient Bed which is used to carry and transfer a patient's body. The most interesting novelty of the approach is that the design is demand-based, i.e. the proposed design of the bed is formulated based on literature survey as well as consult the medical staff. This design approach for the hospital bed goes beyond, and it is based on Latin-American patients, nurse and stretcher-bearers demands. Hospital human resources are the people in daily contact with real situations and needs. For this reason, their feedback is essential to produce a useful hospital bed. This design also plots the basis to consider a functional set of positions demanded by real bed needs. Then, for each required position a mechanism synthesis stage brings a solution for the motion of each required tool. Finally, using tools of mechanical engineering, the complete design can be developed. It is important to mention that the bed construction involves the design and manufacturing of various areas (mechanical, electronic, industrial and graphic design). This integration produces a functional device in combination with an intelligent system [2].

2 Selection of Lifting Mechanism

Lifting machines exist in different configurations. They may be movable, stationary or passive, it depends on their application. Also, this machines may have intermittent action or continuous and may be conducted electrically, by a combustion engine, solar energy or by some other power source. Its principal task is to move a load from one initial point to any other of interest, and they must hold the loads along the path and in the final point. Even that must of the machines are required to move its load strictly in a horizontal motion; some others are designed to develop an angular movement while they are transporting the load. In this work, the Lifting Mechanism must be capable of supporting a weight of 150kg but must be light and strong for greater mobility. As well as being able to develop a desired angular motion.

2.1 Morphologie Analysis

Due to the nature of the design process, there is not a single correct answer for a design problem. Nevertheless, there may be better solutions for the same problem, taking into account the cost-profit. For this reason, it was decided to occupy the Morphology Analysis Methodology [11].

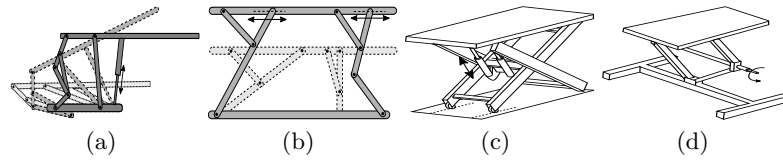


Fig. 1. Four conceptual designs.

This methodology consists of determining multiple conceptual designs while relying on the requirements and characteristics for a system. Following the method, four options were proposed at the beginning of the process. These mechanisms are shown in Fig. 1: a) Lift Storage Mechanism, b) Lift Bench Mechanism, c) Lift Up Mechanism, and d) Scissor Lift Mechanism. In this four possibilities, the link driver to produce the movement of the system is identified with an arrow next to it. In Table 1 is displayed the list of input requirements for the design of the mechanism. This list considers specific behavior and properties of the desired system to establish which one fulfill more of them.

Table 1. Comparison of the options.

Characteristics \ Evaluated mechanisms	a)	b)	c)	d)
Speed				
Few elements				
Few drivers				
No horizontal displacement				
Can vary the slope				
Analysis easiness				
Easy manufacturing				
Low cost				
Stress distribution				

2.2 Final Selection

The four mechanisms were evaluated with the mentioned approach, and the Lift Bench Mechanism, showed in Figure 1(b), had more advantages over the others. The kinematic model was obtained to describe the detailed behavior of each link. The result was a final proposal configuration for this mechanism.

3 Description of Mechanism

3.1 Description of the Lifting Mechanism

The lifting device is composed of two subsystems. They can work together to provide not only the vertical motion of the surface but also the ability to change

the slope of the plane. Figure 2(a) shows a sketch of the entire system with its principal components and actuators to perform the desired behavior. The mechanism one is a four bar mechanism where one of the links is coupled to a linear actuator, marked with the letter A, which changes the position of the slider, thus providing its vertical motion. It is important to mention that one of the links is extended to bond with the mechanism 2. The mechanism two is a five bar mechanism where one of the links is coupled to a linear actuator. In this other case marked with the letter B, this motor changes the position of the slider to produce the vertical motion of this mechanism or to vary the slope of the surface. However, it depends on the relative vertical position of mechanism one to mechanism two. As well as the mechanism one can alter the slope of the platform by changing its relative vertical position to mechanism 2. Figure 1 illustrates that both devices handle the supporting of the whole system. Moreover, working together this two mechanisms can vary the vertical position represented by the variable h in the same illustration. Differential movements from the two actuators produce a slope in the platform of the system measuring the deviation of the normal surface vector from the vertical one. This variable is represented by α . The entire system was selected and configured to produce motion in three distinct configurations. The first one consists of moving the platform surface in strictly vertical position. It means that both mechanisms have the same height and work together to achieve the required horizontal configuration along the vertical displacement of the platform. In the other two motion patterns, both mechanisms collaborate to achieve the desired angle variation in one direction and its opposite.

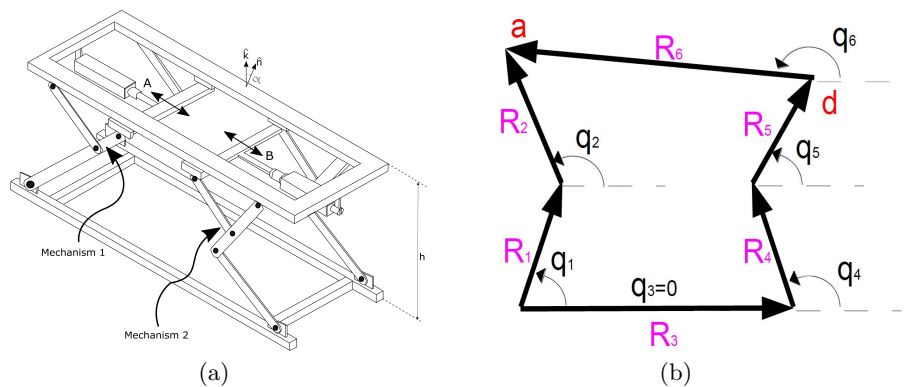


Fig. 2. Physical and schematic description.

3.2 Analysis of Position

A schematic drawing of the lifting machine is shown in Figure 2(b). The entire mechanism can be represented by a six-bar array and it is composed of a reference

bar (R_3), and five links to make a closed chain. The other links have the labels $R_{1,2,4-6}$. On the other hand, $q_i \in \mathbb{R} \forall i = 1 \dots 6$, is the i -th angle between the horizontal axis and the i -th bar with positive counterclockwise direction.

Equation (1) represents the closed chain for the mechanism in Figure 2(b). In this expression each vector is related with each one of the linkages. On the other hand, if vectors are written in polar form [8], (1) can be expressed by equations (2) and (3). Note that when the direction of R_6 is zero, the y -coordinate of the point a and the point d are the same. Therefore, the height of the mechanism can be defined by the position of point a or point d , when the orientation of the vector q_6 is zero. The vertical positions a_y point and d_y are given by equations (4) and (5), respectively. Horizontal positions a_x and d_x are given by equations (6) and (7) respectively.

$$\mathbf{R}_1 + \mathbf{R}_2 = \mathbf{R}_3 + \mathbf{R}_4 + \mathbf{R}_5 + \mathbf{R}_6 \quad (1)$$

$$r_1 \cos(q_1) + r_2 \cos(q_2) = r_3 \cos(q_3) + r_4 \cos(q_4) + r_5 \cos(q_5) + r_6 \cos(q_6) \quad (2)$$

$$r_1 \sin(q_1) + r_2 \sin(q_2) = r_3 \sin(q_3) + r_4 \sin(q_4) + r_5 \sin(q_5) + r_6 \sin(q_6) \quad (3)$$

$$a_y = r_1 \sin(q_1) + r_2 \sin(q_2) \quad (4)$$

$$d_y = r_3 \sin(q_3) + r_4 \sin(q_4) + r_5 \sin(q_5) \quad (5)$$

$$a_x = r_1 \cos(q_1) + r_2 \cos(q_2) \quad (6)$$

$$d_x = r_3 \cos(q_3) + r_4 \cos(q_4) + r_5 \cos(q_5) \quad (7)$$

4 Optimization Problem Statement

The main structure of the system is composed of a six-bar mechanism, and it can be analyzed as a closed kinematic chain. The main objective of this device is to achieve a maximum rise of 90 cm and a minimum height of 40 cm from the base of the mechanism. Likewise, a minimum angle of 16 degrees to the Trendelenburg position is required. The six-bar mechanism design problem is proposed as an optimization problem, where the design variables that optimize the performance function proposal are wanted, subject to constraints inherent in the system. The next subsections mention in detail the design variables, performance function and limitations that are part of the optimization problem design.

4.1 Design Variables

The combination of physical parameters is the responsible for the overall system performance. Based on the requirements already defined, the vector $p \in \mathbb{R}^{12}$ described in (8) which synthesizes the system is proposed. Where the variables r_1-r_6 represents the magnitude of the vectors and the variables q_1-q_6 represents the direction of the corresponding vectors. The elements of this vector physically represent the lengths of the links as well as the angles that each link. The chosen parameters create the set of design variables that can be accessed to modify the system performance.

$$\begin{aligned} p &= [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T \\ &= [r_1, r_2, r_3, r_4, r_5, r_6, q_1, q_2, q_3, q_4, q_5, q_6]^T \end{aligned} \quad (8)$$

4.2 Objective Function

$$\min_x I = w_x(a_x - x_d)^2 + w_y(a_y - y_d)^2 + w_6(q_6 - q_{6d})^2 \quad (9)$$

Where w_x, w_y and w_z are weights in the function.

4.3 Constraints

The constraints imposed to the optimization problem are the maximum-minimum lengths of the links, and the range of feasible angles for each link. These restrictions are expressed in mathematical form in (10)-(14). The constants parameters are defined in Table 2.

$$r_3 \cos(q_3) + r_4 \cos(q_4) + r_5 \cos(q_5) + r_6 \cos(q_6) - r_1 \cos(q_1) - r_2 \cos(q_2) = 0 \quad (10)$$

$$r_3 \sin(q_3) + r_4 \sin(q_4) + r_5 \sin(q_5) + r_6 \sin(q_6) - r_1 \sin(q_1) - r_2 \sin(q_2) = 0 \quad (11)$$

$$r_3 - r_{3d} = 0, \quad r_6 - r_{6d} = 0 \quad (12)$$

$$q_3 = 0, \quad q_6 - q_{6d} = 0 \quad (13)$$

$$\begin{aligned} r_{1min} &\leq r_1 \leq r_{1max} \\ r_{2min} &\leq r_2 \leq r_{2max} \\ r_{4min} &\leq r_4 \leq r_{4max} \\ r_{5min} &\leq r_5 \leq r_{5max} \\ q_{1min} &\leq q_1 \leq q_{1max} \\ q_{2min} &\leq q_2 \leq q_{2max} \\ q_{4min} &\leq q_4 \leq q_{4max} \\ r_{5min} &\leq q_5 \leq q_{5max} \end{aligned} \quad (14)$$

Table 2. Constraints

$r_{1min} = 10[cm]$	$r_{4min} = 10[cm]$	$q_{1min} = 10^\circ$	$q_{4min} = 100^\circ$
$r_{1max} = 50[cm]$	$r_{4max} = 50[cm]$	$q_{1max} = 80^\circ$	$q_{4max} = 170^\circ$
$r_{2min} = 10[cm]$	$r_{5min} = 10[cm]$	$q_{2min} = 100^\circ$	$q_{5min} = 10^\circ$
$r_{2max} = 50[cm]$	$r_{5max} = 50[cm]$	$q_{2min} = 170^\circ$	$q_{5max} = 80^\circ$
$r_{3d} = 150[cm]$	$r_{6d} = 150[cm]$	$w_x = 1$	$w_y = 1$
$w_q = 1$	$x_d = 0[cm]$	$y_d = 90[cm]$	$q_{6d} = 180$

5 Problem Solution and Results

5.1 Sequence of the Solving Process

The optimization toolbox of MatLab was used to solve the nonlinear optimization problem proposed. This tool contains the function `fmincon` that considers the inequality constraints of the optimization problem statement. Due to the nature of the problem, the active-set section of this function was implemented. This approach computes the Hessian approximation needed to obtain the final solution. Figure 3 shows the flowchart of the solving process. On the other hand, this problem does not have a unique solution. Therefore, the next subsections present two different results that strongly depend on the initial condition given to the solving function.

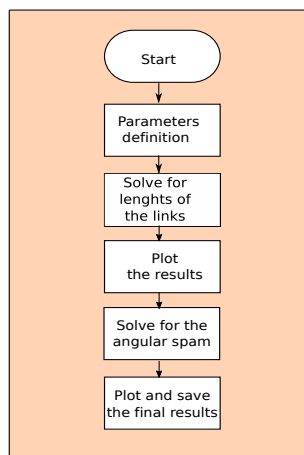


Fig. 3. Optimization flow chart.

5.2 Case 1:

The first case contemplates the initial condition is expressed in (15). The vector solution is in (16), and its components represent the solution for the design

problem. Figure 4 illustrates a view of the final configuration

$$X_{01} = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 180^\circ]^T \quad (15)$$

$$X_{01}^* = [49.98 \ 50 \ 150 \ 49.99 \ 41.95 \ 150 \ 64.15^\circ \ 115.82^\circ \ 0100.81^\circ \ 77.1^\circ \ 179.9^\circ]^T \quad (16)$$

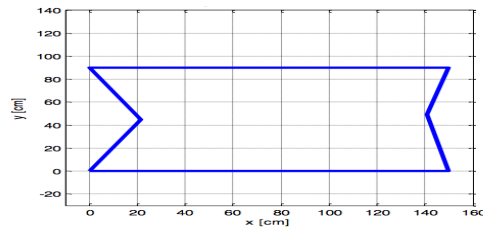


Fig. 4. Configuration for the initial condition 1.

5.3 Case 2:

For this second case, it is proposed the initial condition expressed in (17). The vector solution is in (18), and its components represent the solution for the design problem. Figure 5 illustrates a view of this final configuration.

$$X_{02} = [50 \ 50 \ 150 \ 50 \ 50 \ 150 \ 50^\circ \ 50^\circ \ 0^\circ \ 50^\circ \ 50^\circ \ 180^\circ]^T \quad (17)$$

$$X_{02}^* = [46.74 \ 46.77 \ 150 \ 48.89 \ 44.04 \ 150 \ 74.23^\circ \ 105.74^\circ \ 0^\circ \ 103.69^\circ \ 74.74^\circ \ 179.98^\circ]^T \quad (18)$$

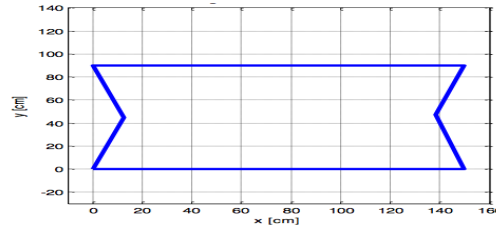


Fig. 5. Configuration mechanism for the initial condition 2.

5.4 Final Results

In sum up, Figure 6 represents the maximum and minimum height for each of the mechanism. This two especial configuration are quite important and they were part of the initial requirements. For this specific plot $q_6 = 80^\circ \pm 16^\circ$.

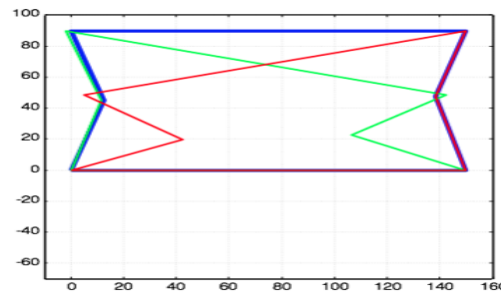


Fig. 6. Complete range for the angular displacement of the platform.

6 3D Modeling

Modeling a system, not only helps to visualize how the prototype would be physically, but also to prevent bad positions, non-realistic pieces, collisions and many others failures. This problem is quite difficult to detect for the designer engineer without the prototyping. One of the biggest advantages when modeling a system is that any detected failure can be repaired in this process. Most of the time this changes involves many other pieces, which can be re-adapted to the design without too many troubles. It is important to mention that the result of the recursive process used many times, showing clearly the importance of using CAD software. In contrast to the traditional design process, which the prototyping is imperative to detect failures and prevent them in the subsequent models, the new technologies provide the advantages of avoiding the cost of prototyping. It leaves everything in the hands of the design engineer. However the recursive process of designing, proving, detecting failures and redesigning,

must be kept. The lifting mechanism modeled in this work is composed of two subsystems. Which provide not only the vertical motion of the surface but also the ability to change the slope of the plane. An isometric view of the entire system is shown in the Figure 7(a), as well as the location of the two subsystems, mentioned before. Finally Figure 7(b) shows the distribution of the load with purple arrows. It contemplates the load that the system will be moving with during its operation. However the details will not be covered in this document.

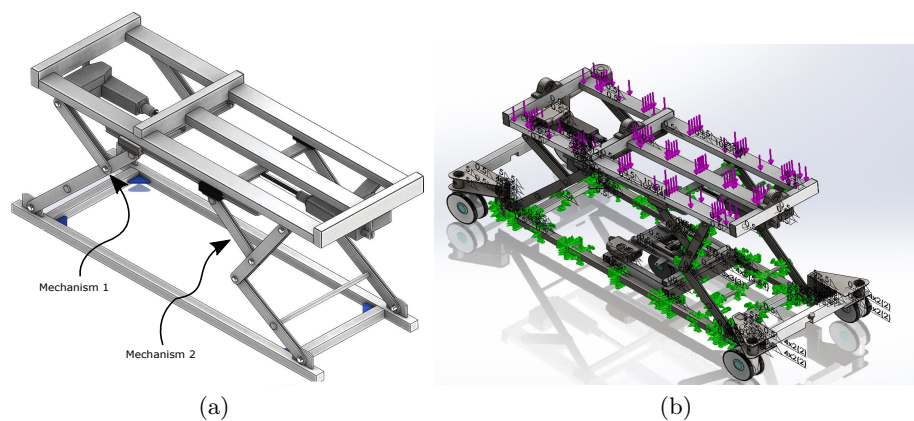


Fig. 7. Physical and schematic description.

7 Conclusion

This paper presented the synthesis of a lifting mechanism as an optimization problem. The analysis of position was carried out and the requirements of the final system were set to guide the process. At the end a complete mechanism was obtained and it is capable of moving according to the restrictions imposed. This work has shown how to take advantage of optimization in order to simplify the design process. Additionally, a 3D model of the lifting mechanism was presented. It has been used 3D CAD software tools for its development. This work contributes, as an example, to the design process, which is currently employed. Despite the traditional process, where most of the time it has to build a prototype, to prove the motion of the system, this work present an alternative methodology with some advantages explained before.

References

1. Andhare, A., Onkar, A., Padole, P.: Design of bed for bedridden patients: Analysis and synthesis of mechanisms. In: 15th National Conference on Machines and Mechanisms, Chennai, Tamilnadu, India. pp. 1–6 (2011)

2. Beltrán-Herrera, A., Vázquez-Santacruz, E., Gamboa-Ziga, M.: Real-time classification of lying bodies by HOG descriptors. In: 6th Mexican Conference on Pattern Recognition MCPR. pp. 211–220 (2014)
3. Ching-Hua, W., Ting-Chun, T., Shin-Chieh, H., Wan-Chun, C., Yen-Ming, C., Kun-Tse, T., Chun-Went, Y., Kuo-YiI, C.: Hospital bed with auxiliary functions of lateral positioning and transferring for immobilized patients. In: The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON). pp. 2991–2995 (2007)
4. Kap-Ho, S., Changmok, O., Tae-Yong, C., Ju-Jang, L.: Bed-type robotic system for the bedridden. In: Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, California, USA. pp. 1170–1175 (2005)
5. Kittipichai, R., Ariyarit, A.: The sizing optimization of hospital bed structure for independently supporting left and or right leg using genetic algorithms. In: International Journal of Modeling and Optimization. vol. 11, pp. 122–188 (2011)
6. Mohammeda, M.N., Khrita, N.G., Abdelgneia, M.A., Abubakera, Es.and Muftaha, A., Omara, M.Z., Salleha, M.S.: A new design of multi-functional portable patient bed. In: Jurnal Teknologi (Sciences & Engineering). vol. 58, pp. 61–66 (2012)
7. Shih-Wei, P., Feng-Li, L., Li-Chen, F.: Mechanism design and mechatronic control of a multifunctional test bed for bedridden healthcare. In: IEEE/ASME transactions on mechatronics. vol. 15, pp. 234–241 (2010)
8. Uicker, J., Pennock, G., Shigley, J.: Theory of Machines and Mechanisms. McGraw-Hill series in mechanical engineering, Oxford University Press (2003)
9. Vazquez-Santacruz, E., Gamboa-Zuniga, M.: A diagnosis methodology for assistive technology development. In: 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). pp. 163–169 (2013)
10. Wang, Y., Butner, S., Darzi, A.: The developing market for medical robotics. Proceedings of the IEEE 9(94), 1763–1771 (2006)
11. Zwicky, F.: Morphologie and Policy Analysis. 16th EURO Conference on Operational Analysis, Brussels (1998)

Programming by Demonstration with DLProlog

José Oscar Olmedo-Aguirre¹, Marisol Vázquez-Tzompantzi²,
Giner Alor-Hernández³

¹ Cinvestav-IPN, Department of Electrical Engineering, Mexico City,
Mexico

² Cinvestav-IPN, DCTS, Mexico City,
Mexico

³ Instituto Tecnológico de Orizaba, Orizaba, Veracruz,
Mexico

oolmedo@cinvestav.mx, galor@itorizaba.edu.mx, mvazquez@cinvestav.mx

Abstract. Traditionally computer programming has been conducted as a long lasting cycle of coding, testing and debugging. In comparison, in programming by demonstration (PBD), a system automatically attempts to construct a program that produces the same input-output pairs given as examples by a demonstrator. Unfortunately, in most PBD systems this automatic program construction process has been approached by finding the proper composition of simpler functions searched from a basic set. The exponential computational costs of such searches makes this approach impractical except for a few domains (i.e. text processing) in which some demonstrator's clues can be used to dramatically reduce the search space. In this work, some of the features of a PBD system under construction are presented that does not rely on function composition but in the step by step construction of a program automaton. The automaton is described in DLProlog, a dynamic logic modal extension of pure Prolog. DLProlog allows to represent the automaton states as user defined first-order predicates and the automaton transitions as logic rules (i.e. dynamic logic modal Horn clauses). The main contributions of this work are twofold: (i) by using a spreadsheet like system for conducting the demonstrations where the basic behavioral rules of the program are learnt by the system, and (ii) by using a dynamic logic modal inference system to construct complex behavioral program rules from simpler ones. Besides the system not only attempts to construct the program but also assists on providing the formal specifications of the input-output conditions, an aspect of paramount importance for program development.

Keywords: Programming by example, logic programming, dynamic logic, knowledge representation and reasoning, machine learning.

1 Introduction

According to [2], in programming by demonstration (PBD), the demonstrator should be able to instruct a system to learn that demonstrator does and the system should create the program that reproduces the users actions. In comparison, in programming by example (PBE) [2, 3], an end-user demonstrates to a system a set of examples consisting of input-output pairs and from them the system will attempt to infer the program that produces these input-output relations. While in PBD the demonstrator provides more guidance to the system to learn the intended program through the observed actions, in PBE the demonstrator only provides a concrete list of input-output pairs from which the system will accomplish the challenging task of deriving the entire program. Hence PBD is intended for programmers with varying degrees of experience, whereas PBE is destined for end-users with no interest in programming the task at hand.

Both programming paradigms are very close areas of research in machine learning, though the more ambitious goals of PBE comes with a price. As argued by Adytia et al. [4], there are two major concerns in this approach: firstly, the very few examples given per task by the demonstrators, and secondly, the searching over arbitrary composition of consistent candidate functions. Possibly, the second one is more challenging because the function composition problem is a NP-complete problem and defies all the state-of-the-art search techniques due to the fact that any change in the input-output relation may radically change the entire program. As an example of the unstable behavior of a PBE system, consider the simple task of reversing lists. As training examples for the PBE system, the demonstrator provides the list of input-output pairs shown in Figure 1:

Input	Output
[]	[]
[a]	[a]
[6]	[6]
[a, b, c, d]	[d, b, c, a]
[0, 1, 2, 3, 4, 5, 6]	[6, 5, 4, 3, 2, 1, 0]

Fig. 1. A list of input-output pairs in the reverse program demonstration.

With no additional information, the PBE system may infer that the output is produced from the input by applying the reverse sorting function. Nonetheless, by including the input [a, b, e, c, d] as testing case, a wrong output is produced as shown next:

Input	Wrong Output	Expected Output
⋮	⋮	⋮
[a, b, e, c, d]	[e, d, c, b, a]	[d, c, e, b, a]

where the reverse sorting function that produced the wrong output needs to be discarded because it does not lead to the expected output. In some problem domains like text processing, the user may provide in the examples additional information that the PBE system uses to draw as many clues as possible. The clues are used in heuristics that may help to reduce the vast search space by selecting better interpretations of the examples. Fortunately, because PBE systems are mainly used in the user-interface, for example of a spreadsheet program, their close interaction with end-users allows the system to acknowledge when the function composed is incorrect whenever the output produced by the system is erased by the user. With this action the PBE system starts searching again for another function that can reproduce the new example along with all the others.

However, the price of this form of unsupervised machine learning of a PBE system is high and not required when the demonstrators are not end-users but programmers. Programmers do not need a system that guess functions that may be wrong for the problem at hand. That programmers need is a programming environment that simplifies the demonstration and where the programmers be able to undertake some abstract reasoning in order to generalize to a parameterized program from the concrete examples provided.

The PBD under design rest on the following principles observed on the programmers behavior when dealing with complex programs:

- Programmers prefer a short cycle of interaction with the input-output pairs obtained by the program. They prefer to find errors as soon as possible in their understanding of the problem rather than spending too much time in a program that may not produce the expected results. Generally, the time spent by programmers to know if their code meet their expectations (not to say specifications) long last in a compiled program than in an interpreted one.
- Programmers find easier to deal with concrete examples instead of going directly through the program codification. Very often programmers need to make some small tests and experiments in order to make sure they understand how to build the program correctly. Unfortunately, the problem with this approach is in the selection of the input data, because they may not consider all the cases in the program design that are required to take into account. However, if the selection exhaustively considers all the relevant cases, this strategy is very helpful to gain deep insights not only on the program design but also on solving the problem at hand.

These principles have motivated the development of a PBD environment that uses DLProlog [8] as its foundational programming language. DLProlog provides the programming concepts used in the environment like program state and program state transition represented by predicates and logic rules (i.e. dynamic logic modal clauses), respectively.

This article is organized as follows. In section 2, a succinct review of the related work is presented. In section 3, a demonstration of how to reverse a finite

list of elements is given. In section 4, the clause learning process is presented in some detail. In section 5, the program construction process is presented as the outcome of an inference procedure. Finally in section 6, some concluding remarks are given along with the future work that already is in progress.

2 Related Work

The history of PBD can be traced back as early as the middle of 1970s with the pioneering work of D. Canfield Smith with his Pygmalion [2] system written in SmallTalk. Pygmalion was intended to change the way of programming in which programs are demonstrated concretely to the computer. One of the main concerns introduced by Canfield is about the human-computer interaction dimensions of a programming language and a programming environment. He argues that writing static language statements interleaved with the computer enactment of such statements becomes a poor way of communication when the translation last long periods of compilation and debugging in opposite directions. Upon this approach of programming, Pygmalion was designed as a graphical programming environment that allows users to sketch their program ideas instead of typing program instructions, in a way that the sketches can actually be executed by a computer. After this pioneering work, a number of research proposals came into the scene [2], like Thinker for programming learning, Rehearsal World for developing educational software, Peridot for creating user interfaces, Chimera for graphical editing, The Geometry Sketch Pad for problem solving in Euclidian Geometry, and TELS for text editing tasks, among others. Though all of them have a notorious influence in modern graphical user interfaces, TELS have silently influenced the text entry systems of our modern intelligent mobile phones.

However, the interest for PBE and PBD spans over fields as different as diverse like Robotics where the purpose is to teach to robots how to achieve some repetitive tasks by user demonstrations [5]. Despite of nowadays there is no a successful and widely used PBE environments, some techniques have found their place and are of common use in widely accepted tools like spreadsheets and graphical user interfaces that learns from user inputs of text and predicts the next input [4]. Because of their theoretical and practical importance, PBE and PBD are important research areas in the AI field of Machine Learning [6].

3 Programming by Demonstration

The PBD system under construction provides a programming environment with a spreadsheet as a front-end user interface along with a small set of basic list processing functions like `hd()`, `tl()` and `cons()`. Assuming that xs denotes a non-empty list, function `hd(xs)` returns the first element of xs and `tl(xs)` returns the remaining list after removing the first element of xs , whereas `cons(x , xs)` written `[$x|xs$]` as in Prolog constructs a list with x as its first element and xs as a list with the remaining elements. The purpose of the environment is to provide an

electronic pad where the programmer can figure out the sequence of operations the program should follow to achieve the task. Figure 2 shows a demonstration of how to reverse a list in the PBD environment.

State	xs	ys_1	ys_2	zs
Reverse ₀	[a, b, c, d]	–	–	–
Reverse ₂	[a, b, c, d]	[]	[a, b, c, d]	–
Reverse ₂	[a, b, c, d]	[a]	[b, c, d]	–
Reverse ₂	[a, b, c, d]	[b, a]	[c, d]	–
Reverse ₂	[a, b, c, d]	[c, b, a]	[d]	–
Reverse ₂	[a, b, c, d]	[d, c, b, a]	[]	–
Reverse ₁	[a, b, c, d]	–	–	[d, c, b, a]

Fig. 2. Demonstration of how to reverse a list.

The demonstration that reverses a list is very simple though illustrative. It proceeds by splitting the input list xs into two lists ys_1 and ys_2 , where ys_1 contains the elements of xs already reversed and ys_2 contains the remaining elements of xs not reversed yet. Thus, for example, the bindings $xs = [a, b, c, d]$, $ys_1 = [b, a]$ and $ys_2 = [c, d]$ describe a valid state during the course of the reverse action on xs . For the next valid state, the first element of ys_2 is removed and inserted at the front of ys_1 . In this way, the list ys_2 of remaining elements is decreased one by one and the list ys_1 of reversed elements is increased. When eventually, the list ys_2 becomes empty, the list ys_1 becomes the reverse of xs . As the functions `app()` and `rev()` need to be defined for the specifications and for the proof of correctness, the several technical aspects in relation to the structural properties of lists cannot be explained here in detail for lack of space. From the usual pure Prolog definitions of `reverse/2` and `append/3`, the definitions of functions `app()` and `rev()` can be established as follows:

$$\begin{aligned} \text{app}(xs, ys) = zs &\Leftarrow \text{append}(xs, ys, zs) \\ \text{rev}(xs) = zs &\Leftarrow \text{reverse}(xs, zs) \end{aligned}$$

From the demonstration, the system will try to identify the minimal set of states that the reversing program requires. By representing states as predicates, the signature of each entry of the table helps to identify each state. The signature is the list of types of the variables that the state holds. Thus the signature of states `Reverse0(xs)` and `Reverse2(xs, ys1, ys2, zs)` are respectively `List` and `(List, List, List, List)`. In Figure 3, the states identified from the demonstration are presented along with their invariant conditions as their definitions.

The invariants are the valid conditions that always hold for the values of the variables bound to them at the corresponding states. Intuitively the invariants ensure that the program only transits among valid states. Therefore, the invariants can be used to state the partial correctness of the program where the initial state `Reverse0(xs)` stands for the precondition and the final state

$$\begin{aligned}
 \text{Reverse}_0(xs) &\Leftarrow \text{List}(xs) \\
 \text{Reverse}_1(xs, zs) &\Leftarrow \text{rev}(zs) = xs \\
 \text{Reverse}_2(xs, ys_1, ys_2) &\Leftarrow \text{app}(\text{rev}(ys_1), ys_2) = xs
 \end{aligned}$$

Fig. 3. States identified from the demonstration.

$\text{Reverse}_1(xs, zs)$ stands for the postcondition. Currently, the demonstrator must provide definitions for the invariants if the user wants to formally proof partial correctness. However, for the derivation of the program, the invariants are not required.

4 DLProlog Clause Learning

The following tables describe the simple rules of computation that the system have learnt from the demonstration. The structure of these tables is similar to the structure of the table used in the demonstration, though restricted to only three rows. The first row shows the name of the rule and the variables with one column for each variable. The second row shows the name of the state at the precondition, the values that each of the variables take at this state, and the guard condition for the rule. The third row shows the name of the state at the postcondition, the values that each of the variables take at this state obtained from the precondition by application of any available functions. For the precondition, the last column shows the guarding condition that selects the valid values to which the rule can be applied. For the postcondition, the last column shows the action, generally an assignment of values taken from the precondition to the variables shown in the postcondition.

In Figure 4, rule R_1 takes the state $\text{Reverse}_0(xs)$ with $xs = [a, b, c, d]$ as precondition and the state $\text{Reverse}_2(xs, ys_1, ys_2)$ with the values $[\]$ and $[a.b.c.d]$ bound to the variables ys_1 and ys_2 , respectively, as postcondition. Because this rule always applies at the beginning of the computation, the program transits unconditionally from state Reverse_0 to state Reverse_2 . In order to infer this rule, at state Reverse_0 the system generalizes the values that can take the input parameter xs to lists with arbitrary but finite number of elements. Then, because at state Reverse_2 the demonstrator puts the empty list into variable ys_1 and copies the values of xs into variable ys_2 , the system infers that the action consists of the multiple assignment that perform this task as shown in the rule. Below the table, the rule R_1 is shown written in DLProlog.

In Figure 5, rule R_2 takes the state $\text{Reverse}_2(xs, ys_1, ys_2)$ with $xs = [a, b, c, d]$, $ys_1 = [d, c, b, a]$ and $ys_2 = [\]$ as precondition and the state $\text{Reverse}_1(xs, zs)$ with xs as before and $zs = [d, c, b, a]$ as postcondition. This rule only applies in the case that $ys_2 = [\]$, when there are no more elements in the list to reverse. In this case, the value assigned to the variable ys_1 is simply copied to the output variable zs . Because Reverse_2 is the final state of the automaton for the reverse program, there are no rules having Reverse_2 as precondition. In consequence there are no more rules that can be applied and then the program stops. Having

$$\begin{array}{l}
 \text{Reverse}_0 \left| \begin{array}{cccc} xs & ys_1 & ys_2 & zs \\ [a, b, c, d] & - & - & - \end{array} \right| \text{true} \\
 \text{Reverse}_2 \left| \begin{array}{cccc} [a, b, c, d] & [] & [a, b, c, d] & - \end{array} \right| (ys_1, ys_2) := ([], xs)
 \end{array} \quad (a)$$

$$[(ys_1, ys_2) := ([], xs)] \text{Reverse}_2(xs, ys_1, ys_2) \Leftarrow \text{Reverse}_0(xs) \quad (b)$$

Fig. 4. Rule R₁ (a) condition-action table, (b) DLProlog clause

the program stopped at this state, the postcondition of the reverse program becomes necessarily satisfied because the postcondition is logical consequence of the state Reverse₂.

$$\begin{array}{l}
 \text{Reverse}_2 \left| \begin{array}{cccc} xs & ys_1 & ys_2 & zs \\ [a, b, c, d] & [d, c, b, a] & [] & - \end{array} \right| ys_2 = [] \\
 \text{Reverse}_1 \left| \begin{array}{cccc} [a, b, c, d] & [d, c, b, a] & [] & [d, c, b, a] \end{array} \right| zs := ys_1
 \end{array} \quad (a)$$

$$[zs := ys_1] \text{Reverse}_1(xs, zs) \Leftarrow ys_2 = [], \text{Reverse}_2(xs, ys_1, ys_2) \quad (b)$$

Fig. 5. Rule R₂ (a) condition-action table, (b) DLProlog clause.

In Figure 6, rule R₃ takes the state Reverse₂(xs, ys_1, ys_2) with $xs = [a, b, c, d]$, $ys_1 = [a]$ and $ys_2 = [b, c, d]$ as precondition and the same state as postcondition although with $ys_1 = [b, a]$ and $ys_2 = [c, d]$. This rule only applies in the case that $ys_2 \neq []$, when there are still some (at least one) elements to be reversed. In the demonstration, there are a number of actions to be taught to the system by: (i) passing the mouse over the list ys_2 that makes selectable both the first and the rest of this list, (ii) selecting the first element of ys_2 , (iii) dragging the selected element to the front of ys_1 and dropping it there, (iii) passing the mouse over the list ys_2 , (iv) selecting the rest of the list ys_2 , and (iii) dragging the selected list to ys_1 and dropping it there. These actions can be summarized in the single DLProlog action $(ys_1, ys_2) := ([hd(ys_2)|ys_1], tl(ys_2))$ that appears at the modality in the head of the clause R₃.

$$\begin{array}{l}
 \text{Reverse}_2 \left| \begin{array}{cccc} xs & ys_1 & ys_2 & zs \\ [a, b, c, d] & [a] & [b, c, d] & - \end{array} \right| ys_2 \neq [] \\
 \text{Reverse}_2 \left| \begin{array}{cccc} [a, b, c, d] & [b, a] & [c, d] & - \end{array} \right| (ys_1, ys_2) := ([hd(ys_2)|ys_1], tl(ys_2))
 \end{array} \quad (a)$$

$$[(ys_1, ys_2) := ([hd(ys_2)|ys_1], tl(ys_2))] \text{Reverse}_2(xs, ys_1, ys_2) \Leftarrow \\
 ys_2 \neq [], \text{Reverse}_2(xs, ys_1, ys_2) \quad (b)$$

Fig. 6. Rule R₃ (a) condition-action table, (b) DL Prolog clause.

DL Prolog rules R_1 , R_2 and R_3 , along with R_0 (not presented here by its simplicity) that were learnt by the system from the reverse program demonstration are ready to be used for the synthesis of a rule with a single compound action that has the behavior expected from the reverse program.

5 Automated Program Construction

The reverse list rules learnt from the demonstration, including also rule R_0 , are shown next:

$$\begin{aligned}
 R_0 &: \text{Reverse}_0(xs) \Leftarrow \text{List}(xs) \\
 R_1 &: [(ys_1, ys_2) := ([], xs)] \text{Reverse}_2(xs, ys_1, ys_2) \Leftarrow \text{Reverse}_0(xs) \\
 R_2 &: [zs := ys_1] \text{Reverse}_1(xs, zs) \Leftarrow ys_2 = [], \text{Reverse}_2(xs, ys_1, ys_2) \\
 R_3 &: [(ys_1, ys_2) := (\text{hd}(ys_2) | ys_1, \text{tl}(ys_2))] \text{Reverse}_2(xs, ys_1, ys_2) \\
 &\quad \Leftarrow ys_2 \neq [], \text{Reverse}_2(xs, ys_1, ys_2)
 \end{aligned}$$

From these rules, the following rule with a complex action can be constructed:

$$R : \left[\begin{array}{l} \text{new } ys_1, ys_2: \\ \left(\begin{array}{l} ys_1, ys_2 := [], xs; \\ \text{while } ys_2 \neq [] \text{ do} \\ \quad ys_1, ys_2 := [\text{hd}(ys_2) | ys_1, \text{tl}(ys_2)] \\ \text{od}; \\ zs := ys_1 \end{array} \right) \end{array} \right] \text{Reverse}_1(xs, zs) \Leftarrow \text{List}(xs)$$

Rule R comprises a single compound action that denotes the intended list reverse program. It can be read as follows: given a list xs , after executing the actions enclosed in the brackets, the output list zs is the reverse of the input list xs whenever xs be a list. The above bidimensional arrangement of the program text (by enlarging the enclosing brackets) is preferred here in order to make it more readable in comparison to its actual linear form:

$$R : [\text{new } ys_1, ys_2: (ys_1, ys_2 := [], xs; \cdots; zs := ys_1)] \text{Reverse}_1(xs, zs) \Leftarrow \text{List}(xs)$$

where the ellipsis indicates the missing program fragment that can be recovered from the previous program presentation by matching the context in which the ellipsis occur. The program synthesis of the reverse program is the outcome of the inference procedure applied to the rules R_0 , R_1 , R_2 and R_3 . The DL Prolog semantic rules for the automated program construction are presented in Figure 7 in the Gentzen's sequent calculus [7].

Note that the DLProlog rules are written in the forward style $F \Rightarrow [A]F'$. The intuitive meaning of the DLProlog rules of Figure 7 can be outlined as the construction of new program rules from others previously constructed, starting with the learnt rules from the demonstration. Each of these rules introduces a construct as described next.

$$\begin{array}{c}
 \frac{\mathbf{P}, F \Rightarrow \neg F' \vdash \mathbf{P}'}{\mathbf{P}, F \Rightarrow [\text{false?}]F' \vdash \mathbf{P}'} \quad (FI) \qquad \frac{\mathbf{P}, F \Rightarrow F' \vdash \mathbf{P}'}{\mathbf{P}, F \Rightarrow [\text{true?}]F' \vdash \mathbf{P}'} \quad (TI) \\
 \\
 \frac{\mathbf{P}, F \Rightarrow F' \vdash \mathbf{P}' \quad \mathbf{P} \vdash \mathbf{P}', G}{\mathbf{P}, F \Rightarrow [G?]F' \vdash \mathbf{P}'} \quad (GI) \qquad \frac{\mathbf{P}, F \Rightarrow F'\{x \mapsto t\} \vdash \mathbf{P}'}{\mathbf{P}, F \Rightarrow [x:=t]F' \vdash \mathbf{P}'} \quad (AI) \\
 \\
 \frac{\mathbf{P}, F \Rightarrow [A]F', F' \Rightarrow [A']F'' \vdash \mathbf{P}'}{\mathbf{P}, F \Rightarrow [A;A']F'' \vdash \mathbf{P}'} \quad (SI) \qquad \frac{\mathbf{P}, F \Rightarrow [A]F', F \Rightarrow [A']F' \vdash \mathbf{P}'}{\mathbf{P}, F \Rightarrow [A \cup A']F' \vdash \mathbf{P}'} \quad (UI) \\
 \\
 \frac{\mathbf{P}, F \Rightarrow F', F' \Rightarrow [A]F', F' \Rightarrow F'' \vdash \mathbf{P}'}{\mathbf{P}, F \Rightarrow [A*]F'' \vdash \mathbf{P}'} \quad (II) \qquad \frac{\mathbf{P}, F \vdash \mathbf{P}', F'}{\mathbf{P}, [A]F \vdash \mathbf{P}', [A]F'} \quad (BI)
 \end{array}$$

Fig. 7. DLProlog semantic rules for program construction.

Rule *(FI)* introduces the modal action `false?` of testing for falsity by removing the negation from the postcondition $\neg F$. It causes to the current course of actions to fail. Rule *(TI)* introduces the modal action `true?` of testing for validity that causes no effect in the course of actions. Rule *(GI)* introduces the modal action `G?` of testing for the guard condition G , whenever G can be deducted. Rule *(AI)* introduces modal action `x := t` of the assignment of term t to variable x , if there is a binding of x to t in the substitution $\{x \mapsto t\}$ applied to postcondition F' . Rule *(SI)* introduces the modal action `A; A'` of sequential composition of simpler actions A and A' , if the postcondition of A implies the precondition of A' . If any of the actions A or A' fails, the sequential composition fails.

Rule *(UI)* introduces the modal action `A ∪ A'` of non-deterministic choice of simpler actions A and A' , if they have both the same precondition F and the same postcondition F' . Its intuitive meaning is that it is possible to take either course of actions given by A or A' because both start at F and both terminate at F' . Besides, if none of the actions fail, one of them is chosen non-deterministically; if any of the actions fails, the other takes place; if both fail, then the non-deterministic choice also fails. Rule *(II)* introduces the modal action `A*` of iteration of simpler action A , if its precondition F implies the invariant condition F' of the iteration and the invariant F' implies the postcondition F'' . If A fails, the iteration terminates.

Finally rule *(BI)* introduces the necessity modal action `[A]` by modal generalization on formula F , meaning that if F is valid in a state it is also valid by any course of actions A that lead to that state. If the action fails, the postcondition becomes necessarily false.

These semantic rules for the program derivation cannot be applied directly because they are expressed in terms of the basic dynamic logic program combinators `false`, `true`, `;`, `∪`, `*`, `:=`. The usual structured constructs of DLProlog like `if then else fi` for selection and `while do od` for iteration must be previously translated into the basic combinators by using the following translation schema:

<code>skip</code>	$\rightarrow \text{true?}$
<code>fail</code>	$\rightarrow \text{false?}$
<code>if F then A fi</code>	$\rightarrow F?;A$
<code>if F then A₁ else A₂ fi</code>	$\rightarrow F_1?;A_1 \cup F_2?;A_2$
<code>while F do A od</code>	$\rightarrow (F?;A) * ; (\neg F?)$

that provides the precise meaning of the usual structured program constructors. Though the semantic rules of DLProlog can be expressed directly using the structured constructors, they are harder to read and understand.

6 Conclusions

In this paper, the foundations of a PBD system are presented. The PBD system works (i) by learning by demonstration the states and the transitions of the program, represented as predicates and rules, respectively, and (ii) by deriving the entire program by recursively assembling simpler rules. The final outcome of the rule construction process is the program with the intended behavior. This approach may prove to be more realistic than function guessing in applications that may not offer any clue.

Among the future work is to finish the user interface and to develop a solid implementation of the semantic rules of DLProlog in order to axiomatically construct correct programs. The integration of both parts is also a hard task to achieve. However, the aim of this research is encouraged by the teaching experience of the first author in the development and verification of non trivial programs. Building an interactive environment that provides the means to teach essential programming concepts is the long-term vision of this work.

References

1. Halbert, Dan: Programming by Example. PhD diss. U.C. Berkeley (1984)
2. Cypher, Allen: Watch What I Do: Programming by Demonstration. MIT Press (1993)
3. Lieberman, Henry: Your Wish is My Command: Programming By Example. Morgan Kaufmann (2001)
4. Aditya, Krishna Menon, et al.: A Machine Learning Framework for Programming by Example. In: Proceedings of the 30th International Conference on Machine Learning, Atlanta Georgia, USA, JMLR: WCP volume 28 (2013)
5. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot Programming by Demonstration. In: Handbook of Robotics, MIT Press (2008)
6. Calinon, S., Guenter, F., Billard, A.: On Learning, Representing, and Generalizing a Task in a Humanoid Robot. In: IEEE Transactions on Systems, Man, and CyberneticsPart B: Cybernetics, Vol. 37, No. 2, pp. 226–298 (2007)
7. Buss, Samuel R.: An introduction to proof theory. In: Samuel R. Buss. Handbook of proof theory. Elsevier, pp. 1-78 (1998)
8. Olmedo-Aguirre, José Oscar, Morales-Luna, G.: A Dynamic-Logic-based Modal Prolog. In: Proceedings MICAI 2012, CPS IEEE Computer Society, pp. 3–9 (2012)

Propuesta de artefactos basados en una notación con grafos y conjuntos para el modelado conceptual de aplicaciones Web

Carlos Daniel Lima Gámez, Silvestre Gustavo Peláez Camarena, Ulises Juárez Martínez,
María Antonieta Abud Figueroa, Celia Romero Torres

Instituto Tecnológico de Orizaba, División de Estudios de Posgrado e Investigación,
Orizaba, Veracruz,
México

{lgcarlosd, ujuarez71}@gmail.com, sgpelaez@yahoo.com.mx,
mabud@itodepi.edu.mx, cromerotres@hotmail.com

Resumen. En el ámbito del desarrollo de aplicaciones Web existen metodologías como UWE (Uml Web Engineering) [1], W2000 [2] entre otras, sin embargo, es práctica común que los desarrolladores omitan su empleo y lleven a cabo de manera más bien empírica que ordenada el desarrollo de las aplicaciones; cuando se utiliza un procedimiento metodológico, comúnmente, los modelos de estas metodologías se construyen basados en el Lenguaje Unificado de Modelado (UML), lo que implica mayor nivel de conocimiento del mismo para la aplicación correcta de las metodologías basadas en éste. Por lo anterior, se proponen en el presente trabajo, una serie de artefactos basados en el uso de grafos y teoría de conjuntos, para el diseño del modelado conceptual de aplicaciones Web, que permitan alcanzar los resultados deseados de manera confiable y expedita por parte del desarrollador.

Palabras clave: Artefactos, aplicaciones Web, grafos, conjuntos.

Proposal of Artifacts based on Notation of Graphs and Sets for Conceptual Modelling of Web Applications

Abstract. In the area of Web application development there are methodologies as UWE (UML Web Engineering) [1] and W2000 [2] among others, however, it is common practice for developers skip their use and perform in empirical way rather than orderly the development of the applications; when a methodology is used, usually its models are built based on the Unified Modelling Language (UML), which implies higher level of knowledge thereof for proper application of methodologies based on it. Therefore, in this paper are proposed a series of artifacts based on the use of graphs and set theory to design the conceptual modeling of Web applications to achieve the desired results reliably and expeditiously by the developer.

Keywords: Artifacts, Web applications, graphs, sets.

1. Introducción

Hoy por hoy, el uso de las aplicaciones Web juega un rol importante como parte de la TI, debido a que día a día las empresas, instituciones educativas y diferentes ámbitos de gobierno ofrecen una amplia variedad de servicios vía Internet. Se puede mencionar como característica importante de las aplicaciones Web, su constante evolución, dado que sus requisitos cambian de manera más rápida que los de una aplicación tradicional y por lo tanto necesitan desarrollarse en menor tiempo. Debido a esta situación, y si los requisitos cambian antes de la liberación de la aplicación, esta podría ya no ser útil y el tiempo y esfuerzo que se dedicó a su desarrollo se perdería.

Una parte importante del modelado de las aplicaciones Web es el modelo navegacional, ya que en él se establecen las rutas que los usuarios pueden seguir en la aplicación para llegar a su objetivo. Por este motivo es necesario un modelo navegacional comprensible.

Otro aspecto relevante, es que las aplicaciones Web tienen características que no se encuentran en una aplicación tradicional, entre las cuales destacan su evolución constante y el aspecto navegacional, razones por las que no pueden tratarse de la misma manera que las aplicaciones tradicionales [3]. Algunas metodologías como UWE [1], W2000 [2] entre otras, basan el desarrollo de sus modelos en la notación de UML, por lo tanto, para aquellos desarrolladores que no tienen el nivel de dominio adecuado de este lenguaje, resulta de difícil comprensión por lo que sus modelos pueden no reflejar lo que en ellos quieren plasmar y por lo tanto al finalizar su aplicación no logran alcanzar las metas pretendidas en el proyecto [6].

Por lo anteriormente descrito, en este trabajo, se proponen artefactos basados en el uso de conjuntos y grafos para el modelado del contenido y la navegación respectivamente, dando como resultado un modelo de clases con una notación con base en la teoría de conjuntos y un modelo navegacional con una notación con símbolos propios de la navegación de una aplicación Web.

2. Trabajos relacionados

En [4] se menciona la poca importancia que se le da al modelado de aplicaciones Web ya que algunos desarrolladores piensan que es pérdida de tiempo y los que llegan a modelar solo lo hacen en las primeras etapas del proceso de desarrollo y después estos modelos no se actualizan con los cambios que se hacen en la etapa de desarrollo. A partir de estos problemas nace la Arquitectura Dirigida por Modelos (MDA) en el cual los modelos evolucionan y se transforman hasta generar de forma automática el código fuente de la aplicación o una parte del mismo. También se plantea el problema de que en la actualidad aunque existen muchas herramientas basadas en la arquitectura MDA que generan código fuente, ninguna de ellas permite generar una aplicación 100% funcional, por tal razón, en [4] se plantea la necesidad de generar una metodología para el modelado de aplicaciones Web que se oriente en la generación del código fuente y que permita a los arquitectos y analistas definir de forma sencilla el comportamiento del sistema. En [5] se menciona el tema de la educación con base en la Web y las variaciones que introdujo en los modelos de la educación formal, ciertos aspectos tales como las comunidades de aprendizaje,

aprendizaje autónomo y colaborativo adquirieron mayor relevancia. Por tal motivo, en [5] se propone una forma de modelar la organización de una comunidad educativa colaborativa en la Web, con el uso de grafos. Para lograr el modelado se describe el esquema propuesto que se basa en una estructura de dígrafo jerárquico el cual contiene un nodo raíz que representa al maestro encargado del curso, se describen con detalle las relaciones que se producen entre los diferentes participantes de la comunidad en el grafo mediante el uso de arcos continuos bidireccionales y arcos unidireccionales. Se concluye que modelar la comunidad no es una tarea sencilla ya que requiere de planificaciones sucesivas conforme evoluciona la comunidad pero gracias a la propuesta del modelado mediante estructuras de grafos resulta alentador ya que para el manejo de estas estructuras existen diversos algoritmos para realizar las planificaciones que se mencionan. En [6] se menciona que el desarrollo de aplicaciones Web es una tarea compleja y desafiante que cuenta con muchas fases y con diferentes tipos de usuarios finales, también se menciona que la mayoría de las aplicaciones Web se desechan o no alcanzan a ganar popularidad y la razón es porque la aplicación no logra las metas de usabilidad o no es lo que los clientes esperaban.

Se afirma que el fracaso de que una aplicación Web no alcance las metas de usabilidad radica principalmente en la pobre comunicación entre los diferentes equipos de desarrollo los cuales generalmente son de diferentes disciplinas (especialistas en interfaces humano-computadora, diseñadores de experiencias de usuario, equipo de aseguramiento de la calidad y equipo de marketing). También se menciona que para facilitar el desarrollo de las aplicaciones Web se necesita un protocolo de comunicación que sea sólido y auto descriptivo para todos los miembros de los diferentes equipos en etapas tempranas del desarrollo. Se propone para tratar de dar solución al problema descrito, un enfoque navegacional con base en bosquejos el cual sea traducible a otros modelos arquitectónicos y a su vez a la posibilidad de generación de código, el diagrama contiene solamente los componentes esenciales que se relacionan con la navegación e interacción del usuario con la aplicación.

3. Artefactos propuestos

Se proponen artefactos para el modelado de contenido y modelo navegacional, como a continuación se detalla:

3.1 Modelado de contenido

El Modelado de contenido se obtiene a partir del modelo de clases definido mediante la notación propuesta con base en la teoría de conjuntos [7, 8] estableciendo como referencia las siguientes reglas:

Regla 1: La letra inicial del nombre de la clase se debe escribir con mayúscula, los atributos y métodos se deben escribir con sus respectivas llaves separados por comas como se establece en la notación de la teoría de conjuntos, por ejemplo:

Persona{ {nombre, apellido, edad},{caminar(), correr()}}

Regla 2: La ausencia de atributos o métodos en la definición de una clase se representa con el símbolo de conjunto vacío " \emptyset ", ejemplo:

Persona{ \emptyset , {caminar(), correr()}}

Regla 3: Para la representación de una clase abstracta se antepone "Abs_" al inicio del nombre de la clase, por ejemplo:

Abs_Persona{{nombre, apellido, edad}, {caminar(), correr()}}

Regla 4: Para representar una interfaz se antepone "In_" al inicio del nombre de la clase, ejemplo:

In_Persona{ \emptyset , {caminar(), correr()}}

Regla 5: Para una clase que implementa una interfaz, se representa utilizando el símbolo de implicación " \Rightarrow ", ejemplo:

Alumno \Rightarrow In_Persona

Regla 6: La herencia se representa utilizando el símbolo " \cup " para unión de conjuntos, ejemplo:

Clase 1: Persona{{nombre, apellido}, {caminar()}}

Clase 2: Alumno{{numctrl, carrera}, {estudiar()}}

"Alumno hereda de persona", se representa como:

Alumno \cup Persona

Regla 7: La composición y agregación de clases, se representan utilizando los símbolos " \in " y " \subset " respectivamente, por ejemplo, para las siguientes clases:

Videoclub{{direccion, telefono}, {agregarPel()}}

Socio{{id, nombre}, {actualizarDatos()}}

SocioVip{ \emptyset , {aplicaDesc()}}

Pelicula{{idp, nombre, genero}, {actualizarInfo()}}

La representación de la composición se muestra en el siguiente ejemplo utilizando el símbolo " \in ":

Pelicula \in Videoclub

El ejemplo a continuación muestra cómo representar la agregación mediante el símbolo " \subset ":

Socios \subset Videoclub

Regla 8: Para establecer la asociación entre clases, utilizar el símbolo de intersección " \cap ", por ejemplo para las siguientes clases:

Busca

Cliente \cap Hotel

La notación anterior significa que existe la asociación entre la clase "Cliente" y la clase "Hotel", la palabra "busca" sobre el símbolo de asociación especifica el tipo de relación que existe entre las dos clases; para el ejemplo se lee "Cliente busca hotel".

Regla 9: Para indicar la multiplicidad (en agregación, composición y asociación), realizarlo como en el siguiente ejemplo:



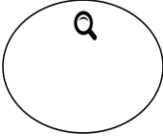

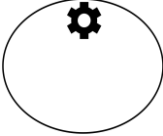

Busca
 Cliente 1..* \cap 1..* Hotel



El ejemplo anterior se lee: "Uno o más clientes buscan uno o más hoteles".

3.2 Modelo navegacional

Los modelos navegacionales muestran las rutas posibles que un usuario puede seguir dentro de la aplicación en función del rol que se le asigne, para realizar este modelo se propone una notación basada en grafos, ésta consta de los elementos que se observan en la Tabla 1.

Tabla 1. Elementos para realizar el modelo navegacional.

Símbolo	Nombre	Descripción
	Nodo Navegacional	Representa un punto de la navegación en el que se le muestra información al usuario.
	Nodo Menú	Representa un menú, es decir, sirve para manejar rutas alternativas de navegación.
	Nodo Consulta	Representa puntos de la navegación donde la aplicación solicita información al usuario.
	Nodo Índice	Representa puntos de navegación donde al usuario se le presenta una lista de posibles resultados para visualizar
	Nodo Proceso	Representa procesos/tareas a realizar en la aplicación
	Acceso Usuario	Sirve para establecer restricciones en la navegación, se le añade un identificador con un rol de usuario y se coloca al inicio de un enlace que sale de cualquiera de los nodos descritos anteriormente para indicar que solamente ese usuario tiene acceso a ese destino de navegación

Símbolo	Nombre	Descripción
	Enlace de navegación	Representa un enlace de navegación de un nodo de navegación a otro, no puede usarse con nodos de proceso/consulta, cuando se omite la punta de flecha se indica que la navegación es bidireccional.
	Enlace de Proceso	Representa un enlace de proceso que va de un nodo de proceso/consulta a otro, no se puede usar para nodos de navegación (para nodos de navegación se usan enlaces de navegación), como en el caso del enlace de navegación si se omite la punta de flecha se indica que la navegación es bidireccional.

4. Primeras pruebas

Para probar los artefactos propuestos, se realizó el modelado de dos casos; el primero consta de una aplicación que muestra los socios de un videoclub, el segundo trata de un sitio de venta de libros.

En la Fig. 1 se muestra el caso de uso del primer caso (Videoclub).

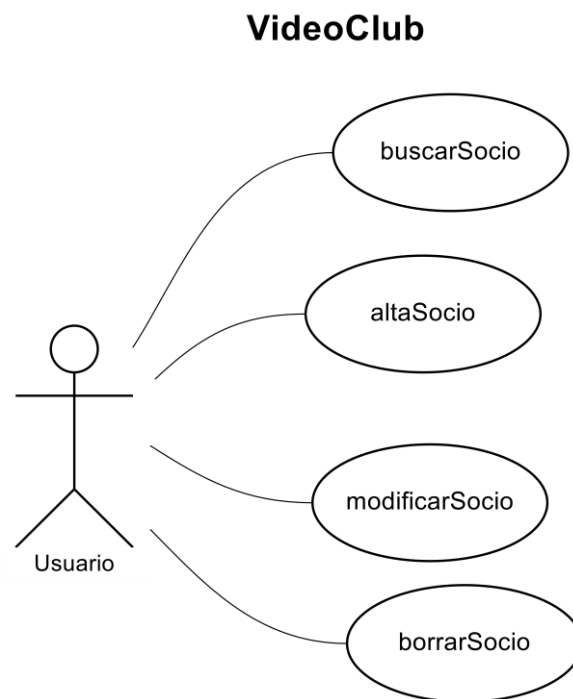


Fig. 1. Caso de uso del videoclub.

A continuación, con base en la notación propuesta de teoría de conjuntos, se desarrolla el modelo de clases de la aplicación como se muestra en la Fig. 2.

Definición de clases	Relaciones entre clases
Videoclub{{descripcion},∅}	Socio 1..* ∈ 1 Videoclub
Socio{{nombre, apellidos, email},∅}	Socio 1 ∩ 1 Vigencia
Vigencia{{inicio, fin },∅}	Socio 1 ∩ 1 Telefono
Telefono{{codigo_area, lada, numero},∅}	Socio 1 ∩ 1 Direccion
Direccion{{dir, cpostal, ciudad, pais},∅}	

Fig. 2. Modelo de clases con base en la notación de la teoría de conjuntos (videoclub).

Como último artefacto para este caso se presenta el modelo navegacional que se obtiene para la aplicación con base en la notación propuesta para este fin (ver Fig. 3).

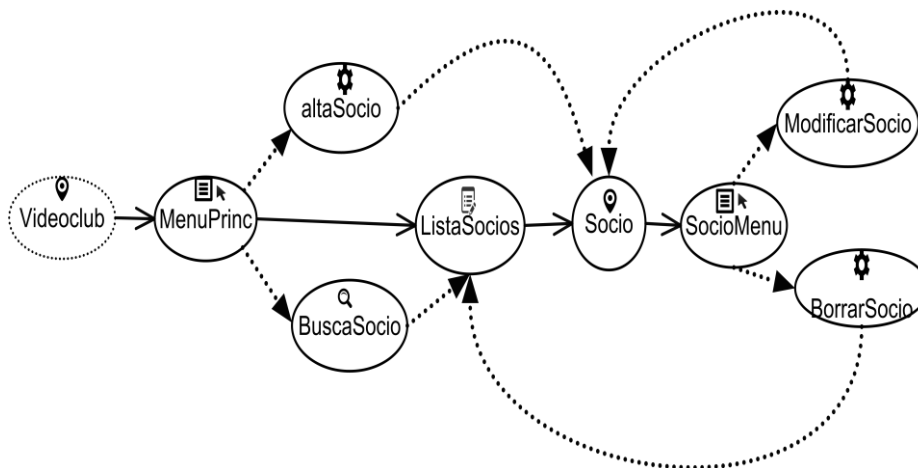


Fig. 3. Modelo navegacional con la notación propuesta (videoclub).

La interpretación del Modelo navegacional anterior es la siguiente: el nodo navegacional con línea discontinua indica el punto de inicio de la navegación, es decir la página principal de la aplicación, el *nodo menú* ("MenuPrinc") muestra tres posibles destinos a seguir ("altaSocio", "ListaSocios", "BuscaSocio"), si se da de alta un nuevo socio ("altaSocio") al finalizar se mostrará la información del nuevo socio, otra opción es ver directamente la lista de todos los socios existentes ("ListaSocios"), pero si se busca por un contacto en específico ("BuscaSocio") se mostrará la lista ("ListaSocios") con los contactos que coincidan con los criterios de búsqueda, cada uno con su respectivo enlace donde se muestra su información a detalle ("Socio"), cada contacto cuenta con dos opciones ("SocioMenu"), la primera para representar la funcionalidad de actualizar un contacto("ModificarSocio") donde al término muestra la información actualizada del contacto y la segunda opción ("BorrarSocio") donde al terminar se regresará nuevamente a la lista de contactos.

A continuación se presentan los artefactos realizados para el segundo caso: en la Fig. 4 se observa el caso de uso para la aplicación "Venta de libros".

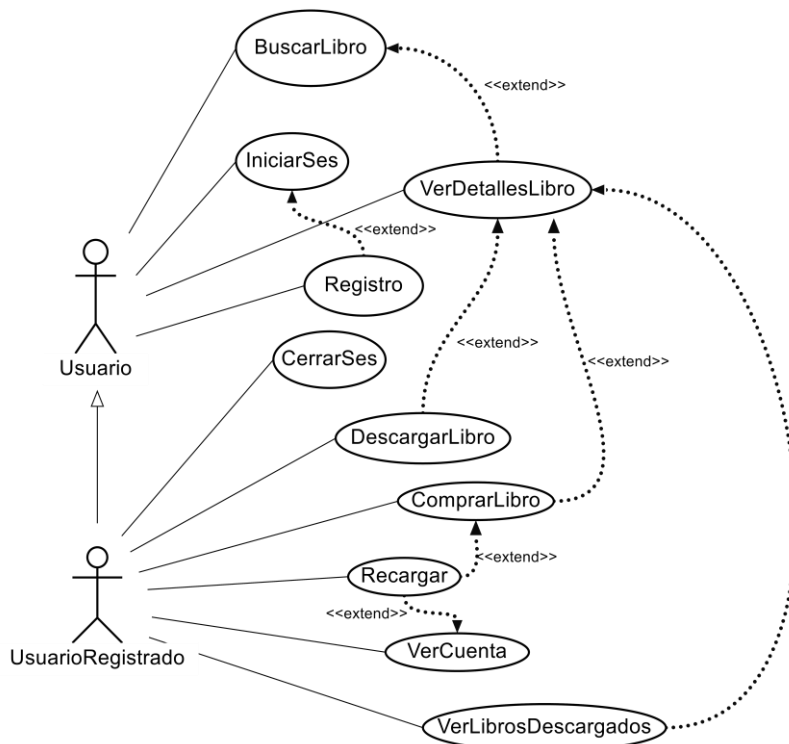


Fig. 4. Caso de uso de sitio de venta de libros.

Posteriormente en la Fig. 5 se observa el modelo de clases para este caso:

Definición de Clases:

Libro {{nombre, precio, linkdescarga, descripcion}, ∅}

Autor {{nombre, pais}, ∅}

Capitulo {{nombre, pagin, pagfin}, ∅}

Usuario {{nombre, contrasenia, creditos}, {comprarAlbum(), recargar(), guardar()}}

Sesion {∅, {establecerSesion(), cerrarSesion()}}

Relaciones:

Sesion 1 ∩ 1 Usuario

Usuario 1 ∩ * Libro

Capitulo 1..* ∩ 1 Libro

Libro * ∩ 1 Autor

Fig. 5. Modelo de clases con base en la notación de la teoría de conjuntos.

Para finalizar se presenta en la Fig. 6 el modelo navegacional del sitio de venta de libros.

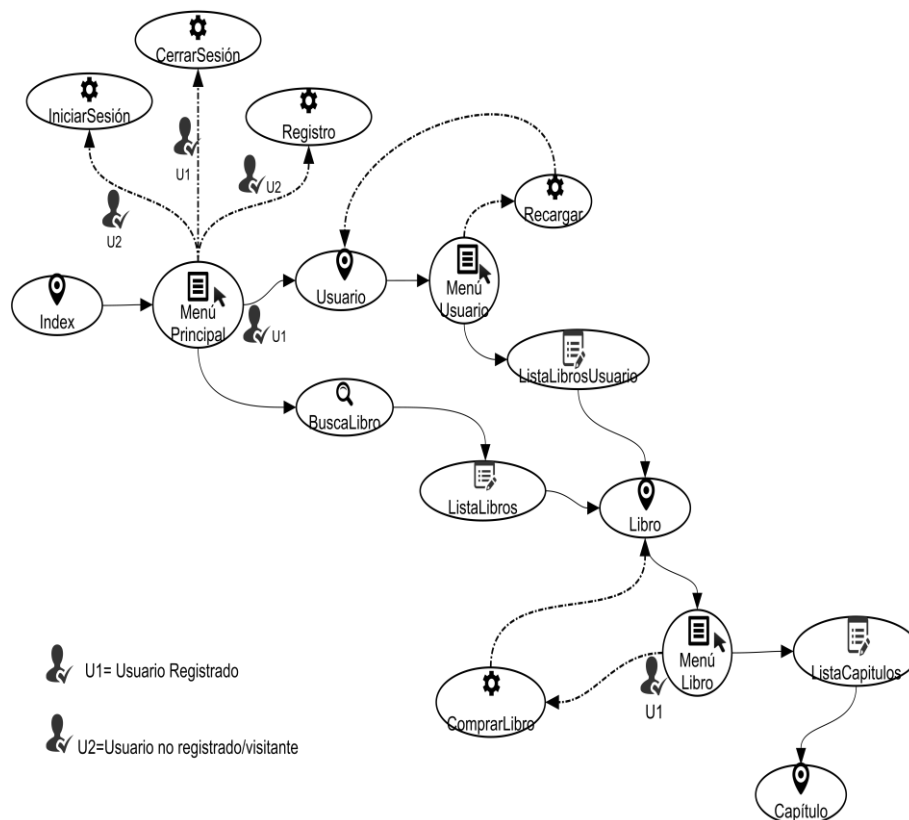


Fig. 6. Modelo de clases con base en la notación de la teoría de conjuntos.

Como se observa en el modelo de la Fig. 6, el elemento que se presenta en la Tabla 1 "Acceso Usuario" ayuda a identificar que usuarios tienen acceso a determinados recursos, como se muestra en el modelo solo el usuario registrado "U1" puede acceder a "CerrarSesión", "Usuario" y "ComprarLibro", con esto se evita el crear un diagrama de navegación para cada usuario que se identifica en la aplicación.

5. Conclusiones y trabajo a futuro

Se puede observar que la teoría de conjuntos cuenta con los elementos necesarios para la representación de un modelo de clases, sin embargo; como todo lo nuevo, es requerido familiarizarse con estos conceptos pero una vez dominada la notación es fácil de aplicar, dando como resultado un esquema más compacto y las relaciones entre las clases se comprenden mejor, además, para la actualización del modelo en caso de que haya cambios a lo largo del desarrollo de una aplicación Web, la notación propuesta resulta flexible para estos casos.

La notación propuesta para el modelo navegacional utiliza símbolos fáciles de relacionar con los aspectos navegacionales de las aplicaciones Web evitando confusiones, por lo que se considera igualmente de fácil aplicación.

Los artefactos propuestos corresponden a la fase inicial para un cambio en el procedimiento metodológico para el modelado de aplicaciones Web. Por lo tanto, como trabajo a futuro se requiere complementar artefactos a este trabajo para la adición del modelo de presentación, y estar en condiciones de posteriormente definir una metodología con base en estos artefactos para así poder realizar una validación formal de los mismos con el desarrollo de un caso de estudio.

Una vez definida la metodología, se considera la implementación de una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) que dé el soporte y así automatizar el uso de los artefactos propuestos.

Referencias

1. Baresi, L., Colazzo, S., Mainetti, L., Morasca, S.: W2000: A modelling notation for complex Web applications. In: Mendes, E., Mosley, N. (Eds): *Web Engineering*, Springer, pp. 335–364 (2006)
2. Koch, N., Knapp, A., Zhang, G., Baumeister, H.: UML-based Web Engineering: An approach based on Standards. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (Eds.): *Web Engineering: Modelling and Implementing Web Applications*, Springer, pp. 157–191 (2008)
3. Sánchez Santamaria, M., García García, L.A.: *La Ingeniería Web: Desarrollo de aplicaciones Web de alta calidad*. HYPATIA, <http://hypatia.morelos.gob.mx> (2011)
4. Vera, P., Pons, C., González, C., Giulianelli, D., Rodríguez, R.: Metodología de Modelado de Aplicaciones Web Móviles Basada en Componentes. In: *XV Workshop de Investigadores en Ciencias de la Computación (WICC)*, pp. 451–455 (2013)
5. Moroni, N., Señas, P.: Uso de grafos para el modelado de experiencias educativas colaborativas basadas en la Web. In: *XIII Congreso Argentino de Ciencias de la Computación (CACIC)*, pp. 1134–1145 (2007)
6. Bouchrika, I., Ait-Oubelli, L., Rabir, A.: Mockup-based navigational diagram for the development of interactive Web applications. In: *Proceedings of the International Conference on Information Systems and Design of Communication (ISDOC)*, pp. 27–32 (2013)
7. Lipschutz, S.: *Sets and basic operations on sets: Set theory and related topics*. McGraw-Hill, pp. 5–6 (1998)
8. Lipschutz, S., Lars Lipson, M.: *Set Theory: Discrete Mathematics*. McGraw-Hill, pp. 1–22 (2007)

ReAHK: Sistema auxiliar para la rehabilitación articular del hombro mediante Kinect

Alan J. Crisantos Bazaldúa, Ángel E. Hernández Beltrán,
Eusebio Ricárdez Vázquez

Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica
Unidad Culhuacan,
Departamento de Ingeniería en Computación, México D.F.,
México

alancris_hotmail.com, aherbel0411@gmail.com, eusebior@ieee.org
<http://www.ic.esimecu.ipn.mx/ALUMNOS/>

Resumen. En el siguiente trabajo se presenta el diseño e implementación de un sistema auxiliar para la rehabilitación articular haciendo uso de KinectTM, el cual consiste en un software no lúdico enfocado en la rehabilitación del hombro con ejercicios que trabajan en el plano frontal y sagital. El software se desarrolló para la plataforma Windows utilizando el lenguaje de programación C# en conjunto con el Kit de Desarrollo de Software MicrosoftTM Kinect. Este sistema supervisa que el paciente realice los ejercicios de forma adecuada y envía un reporte por correo electrónico al médico. El sistema se evaluó con 30 personas sin ninguna lesión de hombro, a quienes se les aplicó un cuestionario de usabilidad después de interactuar con el sistema. Los resultados de la evaluación indican que el sistema es aceptado entre un grupo de personas de diferentes edades como un medio auxiliar para rehabilitación.

Palabras clave: Kinect, sistema para rehabilitación, rehabilitación virtual.

ReAHK: Auxiliar System for Articular Shoulder Rehabilitation Using Kinect

Abstract. This paper presents the design and implementation of an articular auxiliary rehabilitation system based on KinectTM. This is non-ludic software which main target is shoulder rehabilitation based on exercises working in frontal and sagittal plane. This work was developed for the Windows platform using C# programming language and MicrosoftTM Kinect Software Development Kit. This system monitors the patient to perform the exercises properly and email a report to specialist physician. Tests were performed with 30 volunteers without any shoulder injury. After interacting with the system a usability questionnaire was applied. The results of the evaluation indicate that the system is accepted among a group of people of different ages as an aid to rehabilitation.

Keywords: Kinect, rehabilitation system, virtual rehabilitation.

1. Introducción

La rehabilitación virtual ha cobrado importancia en los últimos años, ya que al paso de estos se ha buscado la forma de implementar terapias a distancia para evitar el traslado del paciente a los diferentes centros de rehabilitación [10]. Las terapias físicas que se imparten en un reconocido centro de rehabilitación ubicado al sur de la Ciudad de México, se clasifican en terapia física institucional (requieren de un equipo y/o supervisión especial), terapia ocupacional (ejercicios que se realizan en la vida cotidiana) y terapia física en casa (ejercicios dados por un programa de enseñanza). Los pacientes que realizan sus ejercicios con un programa de enseñanza carecen de un médico especialista que los supervise generando incertidumbre sobre la realización correcta de los ejercicios.

En este trabajo se presenta el desarrollo, diseño e implementación de ReAHK, un sistema auxiliar de Rehabilitación Articular para Hombros con la interfaz Kinect, el cual supervisa la correcta realización de los ejercicios para posteriormente, enviar información sobre el arco de movimiento del paciente mediante correo electrónico y así, evitar posibles lesiones por realizar malas prácticas.

Kapandji [4] define al hombro como una articulación proximal que se encuentra posicionada en la parte superior del cuerpo. Es la articulación con mayor movilidad del cuerpo humano ya que posee tres grados de libertad, es decir, trabaja en los diferentes planos de espacio del cuerpo humano (Plano sagital, frontal o coronal y horizontal o transversal).

Algunos de los ejercicios de rehabilitación que corresponden al hombro son:
a) Flexoextensión que es un ejercicio que consta de dos movimientos, la *flexión* (Figura 1) y la *extensión* (Figura 2), que realizan su movimiento dentro del plano sagital en torno al eje transversal [4].

b) Abducción un movimiento que aleja el miembro superior (brazo) del tronco, partiendo desde la posición anatómica (0°) hasta un ángulo máximo de 180° , este movimiento se realiza en el plano frontal en torno al eje anteroposterior [4], como se muestra en la Figura 3.

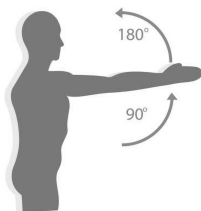


Fig. 1. Ejercicio de flexión.

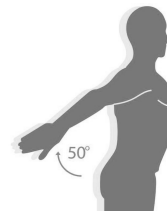


Fig. 2. Ejercicio de extensión.

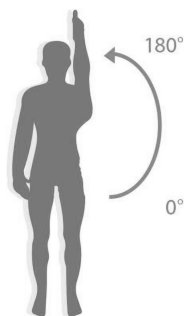


Fig. 3. Ejercicio de abducción.

Para el desarrollo de ReAHK se eligió iniciar con la articulación del hombro a sugerencia de un médico especialista en rehabilitación que consideró a dicha articulación factible de ser rehabilitada usando el Kinect, ya que la abducción y la flexión son ejercicios fáciles de realizar sin supervisión médica.

En este trabajo se describirán los sistemas enfocados en la rehabilitación que utilizan Kinect para la obtención de posicionamiento del usuario mencionando sus principales características, en la siguiente sección se explicarán las técnicas aplicadas para el desarrollo del sistema detallando la manera en que se obtienen los ángulos y cómo éstos son procesados por la máquina de estados, posteriormente, se mencionan los resultados, en donde se describen las características del equipo en el que se realizaron las pruebas. Además, se menciona cómo fue usado y evaluado el sistema y por último, las conclusiones y el trabajo a futuro.

2. Trabajo relacionado

Desde la liberación del kit de desarrollo de software (SDK) de Kinect se han desarrollado diferentes sistemas para la rehabilitación virtual tanto lúdicos (haciendo uso de juegos que sirven para motivar al usuario) como no lúdicos. Sanchis [6] desarrolló un sistema no lúdico basado en Kinect, que permite a los médicos crear guías de los ejercicios que el paciente debe realizar. Mientras que, Tong et al. [8] desarrollaron un sistema lúdico que incluye ejercicios para rehabilitación de pacientes con discapacidades motoras que han sufrido accidente cerebrovascular, con el fin de mejorar su calidad de vida. Por otro lado, González et al. [2] elaboraron un sistema utilizando OpenNI framework y detector de rostros desarrollado en la plataforma *Open Computer Vision Library* para incluirlo en GRADIOR. GRADIOR es una plataforma computacional que ayuda a los médicos en el campo de la evaluación cognitiva y rehabilitación [2]. Este sistema permite llevar un registro de la correcta o incorrecta realización de los ejercicios, los cuales consisten en tocar diferentes características faciales (ojos, nariz y oídos) con las manos. Para obtener las características faciales se procesan las imágenes entregadas por la cámara RGB del Kinect, mientras que

para el seguimiento de la cara y las manos se utilizan los datos de la cámara de profundidad.

En Su et al. [7] se muestra el desarrollo de un sistema de rehabilitación a distancia haciendo uso de Kinect, empleando el algoritmo DTW (Dynamic Time Warping) y lógica difusa. Este sistema utiliza una grabación capturada en el hospital que se usa como referencia, la cual es comparada con el vídeo del paciente usando DTW para medir la similitud entre ambos, dando como resultado la disparidad de trayectoria y la variación de velocidad. Debido a que los médicos se basan en su propia experiencia para evaluar dichas variables hacen uso de la lógica difusa para dar una evaluación subjetiva. De forma similar, Venugopalan et al. [9] proponen un sistema enfocado en la rehabilitación de personas con daño cerebrovascular mediante una sesión de ejercicios empleando Kinect, comparando los datos obtenidos del ejercicio con una grabación de entrenamiento tomada como plantilla. El sistema hace uso de tres técnicas de programación dinámica para evaluar el desempeño del paciente.: *Cross Correlation, Direct Frame by Frame comparison y DTW*.

También existen empresas que han desarrollado sistemas comerciales tales como VirtualWare que desarrolló el software VirtualRehab [3] el cual consta de nueve ejercicios lúdicos que consideran actividades para atender diferentes patologías.

3. Metodología

La idea de desarrollar este sistema nace de las entrevistas que se tuvieron con un médico especialista en rehabilitación que labora en un importante instituto de rehabilitación ubicado al sur de la Ciudad de México, quien mencionó los problemas que puede ocasionar realizar erróneamente un ejercicio y la importancia de mantener una postura correcta mientras se realiza el mismo para que la rehabilitación tenga éxito. Con base en las recomendaciones mencionadas se propuso el desarrollo de un sistema capaz de llevar el control sobre un ejercicio de rehabilitación para hombros (ReAHK), en la Figura 4 se muestra el diagrama de funcionamiento de ReAHK.

A sugerencia del médico, quien indicó que el grado de lesión va ligado a la velocidad de movimiento del brazo, ReAHK no incluye una restricción de tiempo para realizar el ejercicio, con el objetivo de que el paciente pueda avanzar paulatinamente a su propio ritmo sin que se vea obligado a forzar el movimiento de su brazo. Usando los datos de profundidad que transmite el Kinect, es posible llevar un seguimiento en el plano horizontal y limitar el movimiento del brazo de acuerdo al ejercicio con un rango de tolerancia de ± 10 cm con respecto al hombro y la muñeca.

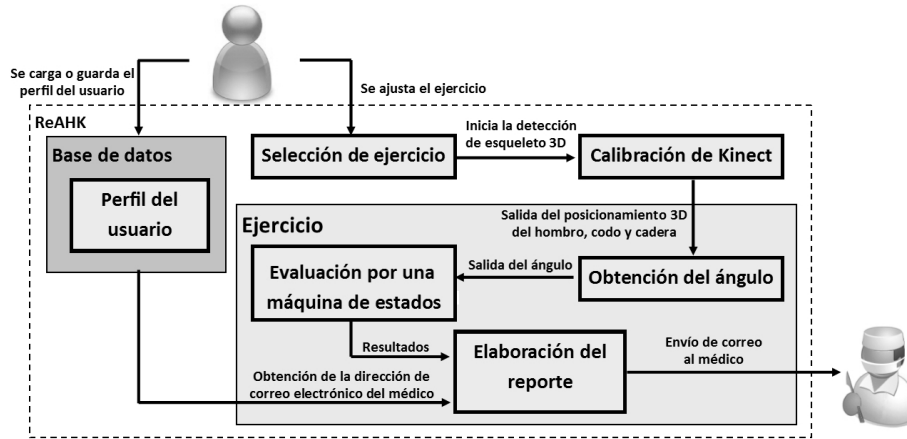


Fig. 4. Diagrama de funcionamiento de ReAHK.

Para poder hacer uso de ReAHK es necesario que el paciente se registre en una base de datos. Una vez iniciado el ejercicio, el sistema captura los movimientos del paciente por medio de las imágenes por segundo que transmite el Kinect (frames), estos contienen la información de posicionamiento de las articulaciones del cuerpo en los tres ejes X , Y y Z , los datos son usados para la obtención del ángulo y validados por una máquina de estados hasta completar la rutina. Al finalizar la actividad, el sistema recupera los datos del perfil del paciente y éstos son usados como plantilla para crear el mensaje de correo que se envía al médico junto con un informe que incluye los ángulos máximos alcanzados en cada repetición y una imagen del paciente mientras realiza el ejercicio.

A continuación, se describirá el método empleado para la obtención de los ángulos que alcanza el paciente durante la realización del ejercicio, la máquina de estados que se utiliza para validar cada repetición y la descripción del sistema completo.

3.1. Obtención del ángulo

Para obtener el ángulo entre las articulaciones (Figura 5) se hizo uso de la expresión analítica del ángulo entre dos vectores (Figura 6), dado que el ángulo entre dos líneas rectas es el más pequeño de los ángulos formados por la intersección de ambas. Únicamente se usaron los ejes X y Y ya que el plano Z es controlado por la cámara de profundidad.

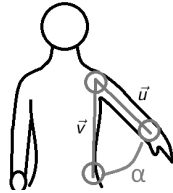


Fig. 5. Obtención del ángulo usando tres articulaciones.

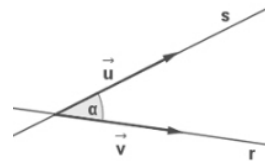


Fig. 6. Representación del ángulo entre dos vectores.

El ángulo entre dos rectas se puede obtener usando sus vectores directores (\vec{u} y \vec{v}) con la fórmula 1 [1,5].

$$\cos \alpha = \frac{|u_1 \cdot v_1 + u_2 \cdot v_2|}{\sqrt{u_1^2 + u_2^2} \cdot \sqrt{v_1^2 + v_2^2}}, \quad (1)$$

donde α es el ángulo formado por la intersección de los vectores \vec{u} y \vec{v} , u_1, u_2, v_1, v_2 corresponden a las coordenadas cartesianas de los vectores.

3.2. Máquina de estados

El ángulo obtenido se valida en la máquina de estados que se muestra en la Figura 7, la cual permite corroborar las repeticiones que realiza el paciente. El ejercicio comienza en la posición inicial (ángulo menor o igual a 15°) y puede tener dos estados de finalización:

Correcto: Se ha llegado al ángulo objetivo (propuesto por el médico) y se ha regresado a la posición inicial.

Incorrecto: No se ha llegado al ángulo objetivo y se ha regresado a la posición inicial.

El estado intermedio, *Ángulo incremental/decremental*, es un estado auxiliar en el que se permanecerá siempre y cuando no se cumplan los estados de finalización mencionados anteriormente.

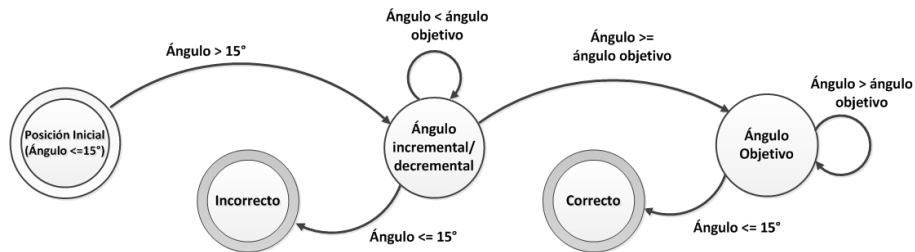


Fig. 7. Máquina de estados para validar una repetición correcta o incorrecta.

3.3. Descripción del sistema

ReAHK está desarrollado en C# utilizando el entorno de desarrollo Visual Studio 2012 en conjunto con el SDK de Kinect versión 1.8. Consta de cinco interfaces: *registro*, *ingreso*, *selección de ejercicio*, *calibración automática* y *ejercicio*. ReAHK al iniciar, despliega una ventana en la que es posible autenticarse con una clave de usuario válida o darse de alta como nuevo usuario. En la ventana de *registro* se solicitan los datos de: nombre del paciente, nombre del médico, usuario, contraseña, correo electrónico del paciente y del médico. En la *selección de ejercicio* (Figura 8) se elige la rutina y se configuran los parámetros del ejercicio como son: la articulación (derecha o izquierda), el número de repeticiones y el ángulo objetivo. La *calibración automática* permite ajustar el sensor para tener una visión completa del paciente con la finalidad de que la captura sea lo más acertada posible. En esta interfaz se implementó el comando de voz “comenzar”, el cual debe pronunciarse de forma fuerte y clara. Al reconocer el comando el sistema inicia la rutina de ejercicio con los parámetros seleccionados. El comando de voz se usa con la finalidad de que el paciente no tenga que moverse de donde esta y no se pierda la calibración obtenida, en esta misma ventana es posible ajustar el ángulo del sensor de forma manual en caso de que la calibración automática falle. También aquí puede apreciarse un vídeo que muestra la realización correcta del ejercicio. Cuando el comando es aceptado se da paso al *ejercicio*, una vista de esta interfaz se presenta en la Figura 9, ésta es el área en la que los pacientes realizan la rutina de rehabilitación, el usuario puede observar en todo momento el ángulo alcanzado en cada repetición y el número de repeticiones realizadas, tanto correctas como incorrectas.

Al completar el número de repeticiones de forma correcta, se despliega la opción para confirmar el envío de información al correo del médico. El reporte que se envía incluye el número de repeticiones y el ángulo obtenido en cada una de ellas, además se adjunta una foto capturada mientras el paciente realizaba el ejercicio, con la finalidad de que el médico observe la postura del paciente.



Fig. 8. Selección del ejercicio.

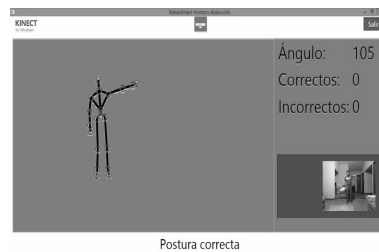


Fig. 9. Realización del ejercicio.

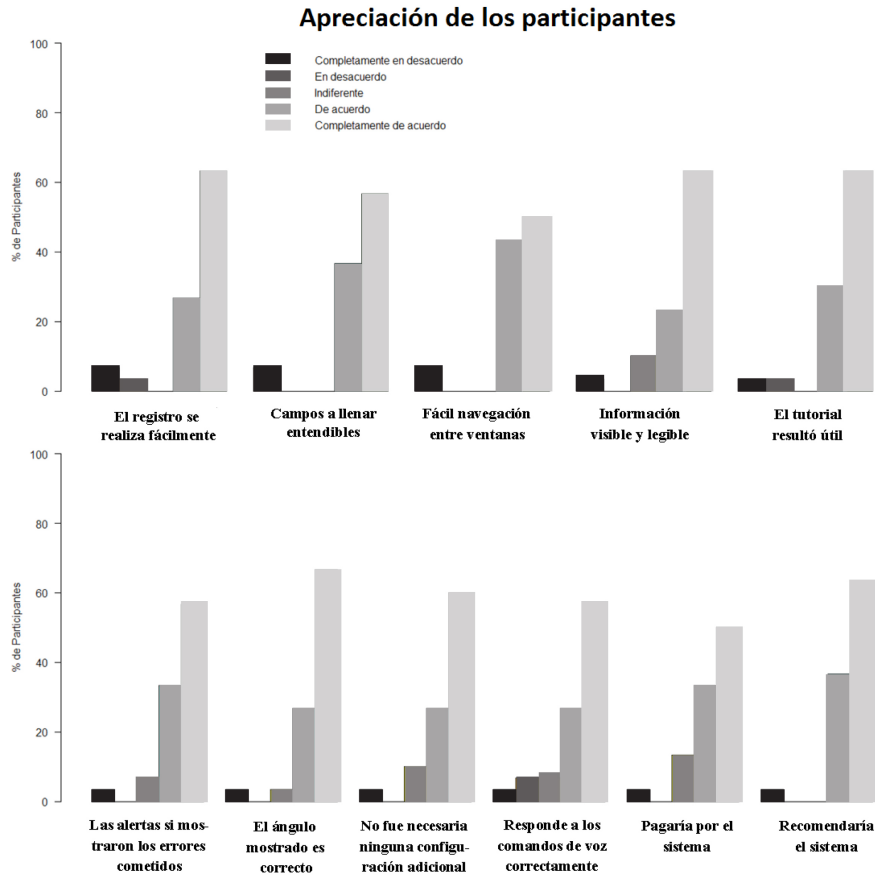


Fig. 10. Resultados del cuestionario de usabilidad.

4. Resultados de usabilidad

Para evaluar la usabilidad del sistema, se realizaron pruebas con 30 personas sanas (sin ninguna lesión en el hombro) con un rango de edad de 18 a 58 años. Las pruebas se realizaron utilizando una computadora portátil marca MSI modelo CX61 con procesador i7-3630QM a 2.4GHz y memoria RAM de 6GB DDR3 además del dispositivo Kinect para Xbox 360.

Se le proporcionó al usuario una breve introducción al sistema en la que se explicó el funcionamiento de cada interfaz y en qué consistían los dos ejercicios implementados. Se procedió a que cada uno de los participantes realizara el proceso de registro y un ejercicio de rehabilitación (abducción o flexión). Una vez completada la rutina, cada participante llenó un cuestionario de usabilidad en el que se calificaron tres aspectos: diseño, funcionalidad y opinión general.

En la Figura 10 se puede apreciar que las calificaciones más altas fueron en las preguntas correspondientes al diseño y funcionalidad. Los usuarios que calificaron de manera negativa manifestaron que se confundían en el registro o la configuración de cada interfaz principalmente en personas mayores de 40 años.

Así mismo, en la Figura 10 se observa que el 65% de los usuarios consideran que los ángulos mostrados en pantalla corresponden al ejercicio real que se está realizando. Además, más del 60% de los mismos, recomendarían el sistema a personas con problemas en el hombro. Sin embargo, las principales valoraciones negativas se presentan en el comando de voz, lo cual se puede atribuir al ruido y a la sensibilidad del micrófono.

De los voluntarios que realizaron la prueba y contestaron el cuestionario, nueve manifestaron haber necesitado rehabilitación en el pasado y consideraron que el sistema puede ser de mucha utilidad ya que evitaría el deterioro del hombro por realizar malas prácticas.

5. Conclusiones y trabajo a futuro

Las ciencias computacionales pueden apoyar a diferentes campos de estudio. En este proyecto se adaptaron algunas herramientas, tanto de software como de hardware, implementándolas en la rama de medicina física y rehabilitación, dando como resultado un sistema que facilita al médico el seguimiento del progreso de personas que realizan su terapia de rehabilitación a distancia al tener un registro de los reportes de los ángulos alcanzado por el paciente durante toda su terapia.

Analizando los resultados del cuestionario de usabilidad y tomando en cuenta a los voluntarios que han necesitado rehabilitación en el pasado, consideramos que el sistema puede ser de gran utilidad para apoyar a las personas que estén realizando ejercicios de rehabilitación sin supervisión.

Previo a la aplicación de las pruebas, el sistema fue validado por el médico especialista, quien consideró que con los resultados obtenidos es posible implementar el sistema con pacientes que sufran de alguna lesión del hombro y necesiten rehabilitación (ejercicios de flexión y abducción).

Con base en las opiniones recabadas durante el periodo de pruebas realizadas, algunas de las mejoras que se pueden realizar al sistema son: ampliar el software agregando más ejercicios que permitan rehabilitar diferentes articulaciones como el codo y la rodilla, además de implementar más ejercicios para el hombro. También, se puede mejorar la interfaz gráfica pasando del diseño 2D a un diseño 3D, para que el usuario le sea más fácil comprender los movimientos que realiza.

Agradecimientos. Agradecemos al Dr. Marco Antonio Guerrero Godínez, médico especialista en rehabilitación por su valiosa asesoría y apoyo.

Referencias

1. Bo, A.P.L., Hayashibe, M., Poignet, P.: Joint angle estimation in rehabilitation with inertial sensors and its integration with Kinect. In: Proceedings of the Annual

- International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS). pp. 3479–3483 (2011)
2. González-Ortega, D., Díaz-Pernas, F.J., Martínez-Zarzuela, M., Antón-Rodríguez, M.: A Kinect-based system for cognitive rehabilitation exercises monitoring. *Computer Methods and Programs in Biomedicine* 113(2), 620–631 (2014), <http://dx.doi.org/10.1016/j.cmpb.2013.10.014>
 3. Group, V.: VirtualRehab (2015), <http://www.virtualrehab.info/es>
 4. Kapandji, A.I.: *Fisiología Articular: Tomo 1*. Editorial Medica Panamericana, España, sexta edn. (2006)
 5. Ochoa, C.O., Valencia, J.V., Cordova, J.L., López, A.O., Ortíz, M.M.: Análisis de marcha mediante plataforma Kinect. pp. 17–21 (2011)
 6. Sanchis Cano, A.: *MediKnect: Un sistema de rehabilitación virtual*. Ph.D. thesis, Escuela Técnica Superior de Ingenieros de Telecomunicación (2013)
 7. Su, C.J., Chiang, C.Y., Huang, J.Y.: Kinect-enabled home-based rehabilitation system using Dynamic Time Warping and fuzzy logic. *Applied Soft Computing* 22(November 2010), 652–666 (2014), <http://linkinghub.elsevier.com/retrieve/pii/S1568494614001859>
 8. Tong, R.K., Hang, C.H., Chong, L.K.W., Lam, N.K.F.: KineLabs 3D motion software platform using Kinect. In: *International Conference on Computerized Healthcare (ICCH)*. pp. 164–165 (2012)
 9. Venugopalan, J., Cheng, C., Stokes, T.H., Wang, M.D.: Kinect-based rehabilitation system for patients with traumatic brain injury. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference 2013*, 4625–8 (2013), <http://www.ncbi.nlm.nih.gov/pubmed/24110765>
 10. Yunda, L., Gómez Ortega, L.F., Rodríguez, S., Millán, R.: Plataforma Web para un nuevo modelo de tele-rehabilitación, de base comunal, en áreas rurales. *Sistemas y Telemática* 9(19), 55–67 (2011), http://aplicaciones.icesi.edu.co/revistas/index.php/sistemas/_telematica/article/view/1094

Reconocimiento de patrones numéricos para vuelo controlado de un AR drone utilizando redes neuronales artificiales

Juan Carlos Rodríguez-Sánchez, Víctor Manuel Landassuri-Moreno, José Martín Flores Albino

Universidad Autónoma del Estado de México, Centro Universitario UAEM Valle de México
México

{jcrsylatp, jose_martin_70}@yahoo.com, vmlandassurim@uaemex.mx,

Resumen. En el presente trabajo se utilizó la segmentación de letras de fotografías tomadas mediante la cámara del Parrot AR Drone, con el objetivo de establecer una interacción de estímulo-respuesta, donde la imagen original en formato de combinación de colores Rojo, Verde y Azul (RGB) se segmentó por color (escogiendo el canal rojo). Una vez que se reconoce el carácter, el Drone ejecuta la acción correspondiente. Se utilizaron inicialmente patrones numéricos libres de ruido y posteriormente se agregaron algunos pixeles a la imagen con el objetivo de hacer más robusto este conjunto de patrones, los cuales proporcionaron el conjunto de entrenamiento para la red neuronal y de esta forma se pudieron interpolar patrones nuevos. Para la segmentación de imágenes se utilizaron técnicas de detección de bordes que incluyen el filtro de Sobel así como filtros para eliminación de ruido basados en filtrado de la mediana que es un filtro pasa baja. Todo lo anterior se llevó a cabo en un entorno cerrado y se espera ampliar este trabajo para su aplicación en diferentes entornos.

Palabras clave: Segmentación de caracteres, vehículos aéreos no tripulados (UAVs), procesamiento de imágenes, filtro Sobel, detección de bordes, redes neuronales artificiales.

Recognition of Numeric Patterns for a Flight Controlled by an AR Drone Using Artificial Neural Networks

Abstract. In this paper the letter segmentation of photographs was used, taken from a Parrot AR Drone's camera with the aim of establishing a stimulus-response, where the original picture formed by Red, Green and Blue (RGB) colors was segmented by color (choosing the red channel). Once the character is recognized, the Drone executes the corresponding action. Noise-free number patterns were initially used and then some pixels were added in the image in order to make a set of patterns more robust, which provided the training set for neural network and thus are able to interpolate new patterns. Edge techniques detection were used for image segmentation including Sobel filter and filters for noise removal based on the median filtering, that is a low pass filter. All this took place in a closed environment, expecting to extend this to different environments.

Keywords: Character segmentation, unmanned aerial vehicles, image processing, Sobel filter, border detection, artificial neural networks.

1. Introducción

El ojo de los mamíferos es un órgano complejo capaz de procesar señales luminosas que el cerebro interpreta para llevar a cabo diversas tareas donde, en el caso particular del ojo humano, éste es capaz de detectar figuras a diferentes distancias, con diferentes rotaciones y niveles de ruido. De esta forma en un lugar donde se proyecta sombra, aún es posible reconocer formas e incluso colores. Por otro lado, hasta hace poco el reconocimiento e interpretación de caracteres escritos tales como letras y números resultaba ser una tarea de gran complejidad para una computadora, ya que ésta solo procesa señales digitales. Por otra parte los colores corresponden a diferentes frecuencias del espectro electromagnético, las cuales son señales continuas en el tiempo, i.e. son analógicas y por consiguiente requieren un procesamiento previo antes de ser manipuladas por medio de procesamiento digital. A pesar de ello, el desarrollo de diversas herramientas de software ha facilitado considerablemente las tareas de procesamiento de imagen, donde estos programas han visto mejorado su rendimiento mediante la incorporación de las Redes Neuronales Artificiales (RNAs).

En lo referente al control de los parámetros de vuelo de los UAVs explicados a detalle en [1, 2, 3], se requiere de un control adecuado y las RNAs en combinación con las técnicas de procesamiento digital de imagen proporcionan una herramienta eficiente para este propósito. Una amplia revisión de aplicaciones de RNAs para procesamiento digital de imágenes se menciona en [4] donde además se muestran las etapas de nivel de procesamiento que son: pre-procesamiento, Segmentación, Detección, Clasificación y Análisis. Otras técnicas involucran el uso de estadísticas para el análisis minimizando el riesgo de hacer una clasificación incorrecta de los patrones [5] o mediante técnicas matemáticas de rechazo de umbrales con aprendizaje no supervisado [6]. Lo anterior demuestra que las redes neuronales resultan ser muy útiles al momento de trabajar con imágenes.

Por otra parte, el uso cada vez más frecuente de robots de servicio aplicados en diferentes áreas ha motivado el interés por el mejoramiento de la interacción Hombre-Máquina y por ello se han llevado a cabo algunos experimentos de reconocimiento de patrones utilizando robots terrestres en entornos cerrados, utilizando odometría y codificadores ópticos como se señala en [7] y otros analizando la interacción con las máquinas en entornos cambiantes [8]. Así, la adquisición de imágenes mediante cámaras conectadas a robots, busca establecer una relación de estímulo-respuesta procesando comandos para ejecutar acciones concretas en entornos reales, apoyándose a su vez en simulaciones 3D virtuales, como lo señalado por [9]. En el caso particular de los vehículos aéreos no tripulados (UAVs), algunas técnicas involucran la elaboración de mapas vectorizados basados en imágenes tomadas vía satélite mediante la detección de contornos y objetos [10], mientras que otros analizan los resultados de la interacción del AR Drone mediante el diseño de una interfaz para el reconocimiento de órdenes gestuales [11]. Un trabajo similar se muestra en [12], donde ya se utiliza el AR Drone para aterrizaje y despegue autónomo, implementando

técnicas de cotejamiento de patrones buscando pequeñas secciones dentro de una imagen o mediante técnicas de detección de aspectos y cotejamiento que basa su funcionamiento en la descripción de puntos mediante vectores utilizando sensores de ultrasonido instalados en el Parrot AR Drone. Así, éste artículo presenta la “segmentación de caracteres de fotografías tomadas mediante la cámara del Parrot AR Drone”, los algoritmos programados se basaron en el uso de operadores morfológicos, todos éstos enfocados al pre-procesamiento de la imagen. Se generaron patrones a partir de los caracteres segmentados con distintos niveles de ruido, mediante el uso de RNAs con un algoritmo evolutivo para adaptar sus arquitecturas durante evolución.

Este trabajo está organizado en seis secciones. La primera sección corresponde a los antecedentes. La segunda sección corresponde a la descripción de los métodos propuestos, posteriormente en la cuarta sección se muestra la configuración experimental, los resultados obtenidos se muestran en la sección cinco. Finalmente en la sección seis se describen las conclusiones y trabajos futuros.

2. Antecedentes

En esta sección se describirán las técnicas utilizadas para aislar el carácter del resto de la imagen. En primer lugar se tienen los llamados métodos de filtrado que permiten resaltar o suprimir de forma selectiva con la finalidad de destacar algunos elementos que forman parte de una imagen. Algunos filtros de imagen son los llamados filtros pasa baja que incluyen el filtro de la media, el filtro de la media ponderada, el filtro de la mediana, filtros adaptativos y filtros Gaussianos. También se tienen los llamados filtros pasa alta que incluyen la sustracción de la media, los filtros basados en la derivada y los filtros Laplacianos. El método utilizado en este trabajo es el de filtrado de la mediana el cual aplica una matriz de 3x3 a cada uno de los pixeles en una imagen en escala de grises, la cual funciona a partir de dos máscaras, una horizontal en función de “x” y otra vertical en función de “y” (figura 1).

1	1	1
1	1	1
1	1	1

DIV = 9

Fig. 1. Filtro de la media.

Existen otros métodos que consideran que los bordes de la imagen no podrán procesarse debido a que se excederían los bordes de la imagen perdiendo N-2 Filas X N-2 Columnas.

3. Métodos propuestos

El software National Instruments - LabVIEW proporciona un método adecuado de trabajo para el desarrollo de instrumentación virtual que posee las capacidades de manipular hardware desde una tableta, computadora o teléfono celular. Debido a su flexibilidad en lo que se refiere al procesamiento de imagen IMAQ, es una herramienta muy utilizada en el ámbito industrial, tal y como se muestra en [13]. En una primera etapa se diseñaron interfaces para control de vuelo utilizando el LVH AR Drone Toolkit, posteriormente se enlazaron las cámaras del AR Drone con la cámara de una tableta. Esto permitió tomar fotografías de los caracteres impresos con la cámara del Drone, para lo cual se utilizaron herramientas de LabVIEW. Este procesamiento incluye herramientas basadas en el toolbox IMAQ dentro de la opción Vision and Motion/Vision Utilities. Las herramientas de este toolbox incluyen al IMAQ Create para reservar un espacio en memoria para la imagen. Dentro de la opción Vision and Motion se selecciona NI-IMAQdx y una vez ahí las herramientas Open, Configure Grab y Close se encargan de abrir la imagen, mantenerla en memoria y cerrar la sesión. Además la herramienta “Vision Acquisition” permite registrar los bits de inicio en la entrada. Posteriormente se almacenaron las imágenes adquiridas en una carpeta convirtiendo los valores RGB de la imagen de entrada a valores numéricos enteros. La figura 2 muestra un código para capturar la imagen con el AR Drone.

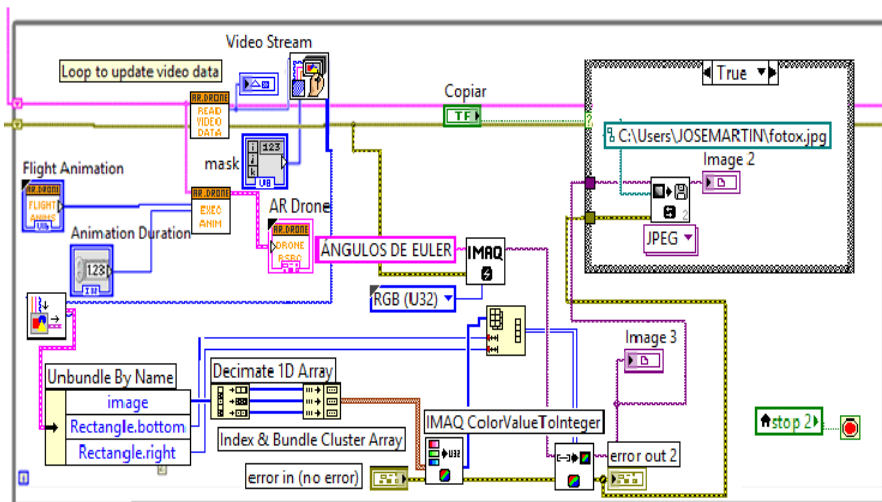


Fig. 2. Código para captura de Imagen.

En la siguiente etapa del desarrollo se utilizó Matlab para llevar a cabo la segmentación de los caracteres y su posterior reconocimiento y procesamiento. Con el propósito de eliminar la mayor cantidad de ruido posible de las fotografías tomadas con la cámara del AR Drone, los números se imprimieron en color Rojo por lo que en el algoritmo utilizado se separa la imagen RGB en sus correspondientes canales Rojo, Verde y Azul para eliminar el fondo y toda la información innecesaria. Esto se muestra en la figura 3.

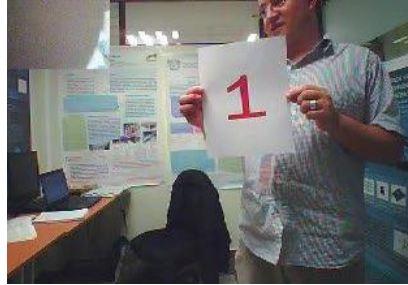


Fig. 3. Imagen Original con un comando numérico.

Aún es necesario quitar ciertas regiones que están formadas por diferentes grupos de píxeles por lo que fue necesario aplicar el filtro de la mediana y el filtro de imagen binaria. De esta forma se obtiene una imagen con una menor cantidad de píxeles, reduciendo considerablemente el ruido, tal y como se puede apreciar en la Figura 4.

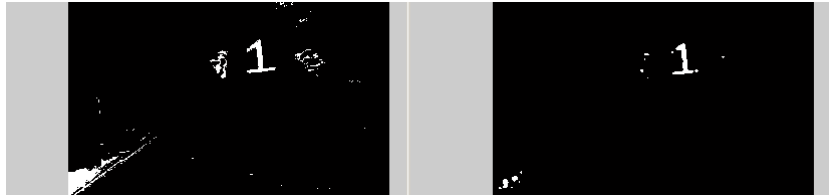


Fig. 4. Imagen con Ruido (izquierda) e imagen con filtro de la mediana y filtro de imagen binaria (derecha).

Algunos grupos de píxeles se pueden apreciar en la imagen por lo que se utilizaron funciones que permiten la eliminación de grupos de píxeles menores a un umbral cuyo valor en este caso se fijó a 100 (Figura 5).

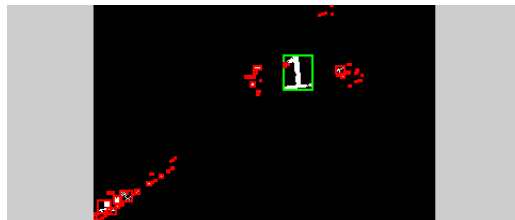


Fig. 5. Eliminación de los grupos de píxeles con un valor menor al umbral.

En la figura 6 se muestra el resultado final de la segmentación.



Fig. 6. Resultado final de la segmentación.

Para generar los patrones que servirán como datos de entrada para la red neuronal se rotaron las imágenes a 45° cada una con respecto de la anterior hasta completar los 360° . En la siguiente imagen se muestra la secuencia de rotación para el número 1.



Fig. 7. Secuencia de rotaciones.

Considerando todos los números del 1 al 8 se obtuvo este primer conjunto de patrones que consta de un total de 64 patrones. Una vez que se tienen estos datos se separa la imagen en sus columnas y se concatenan de forma horizontal con la etiqueta que servirá para tener un entrenamiento supervisado con las redes neuronales. Esta etiqueta es un vector de bits el cual sirve como identificador para cada patrón. En la figura 8 se puede ver la imagen antes y después de agregar el identificador, cabe destacar que la imagen final binarizada fue concatenada con todas sus filas, así como con la etiqueta, dando un tren de bits con el identificador concatenado al final, como se muestra en la figura 8.

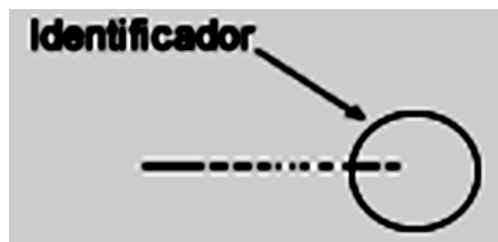


Fig. 8. Se añade el identificador a la secuencia de bits.

Un conjunto de patrones ya concatenados se muestra en la figura 9, donde cada línea representa un patrón de entrada para la red neuronal. Ahí es posible apreciar que las últimas columnas de la Figura 9 son iguales, ya que todas corresponden al dígito 1.



Fig. 9. Bits concatenados para el patrón de 1.

Con el objetivo de lograr un conjunto de entradas para la red neuronal más robusta, se generó ruido aleatorio en cada uno de estos patrones. En la figura 10 se muestra uno de los patrones girado y con ruido. La figura 11 muestra en algoritmo de flujo del programa implementado en este trabajo.

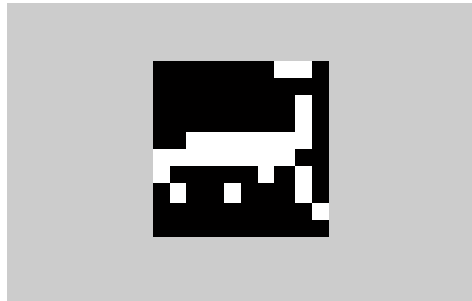


Fig. 10. Ejemplo de Imagen con ruido.

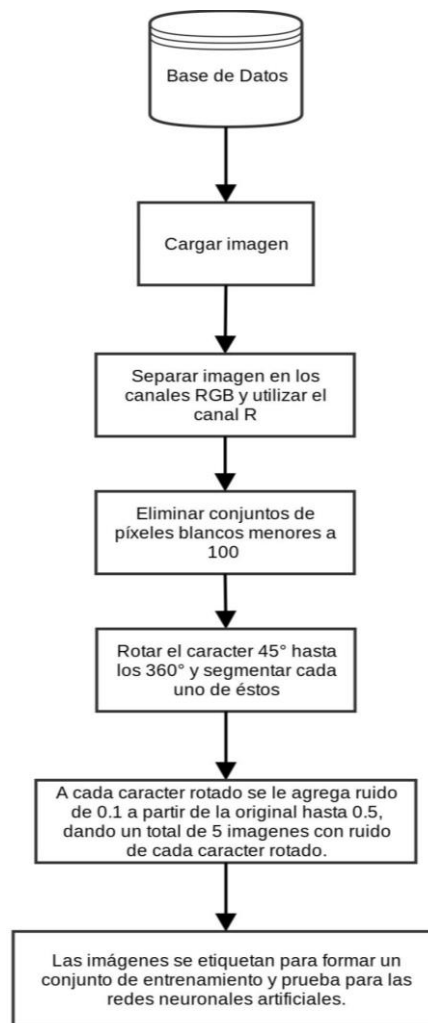


Fig. 11. Diagrama de Flujo del algoritmo.

4. Configuración experimental

Cada patrón (dígito) fue girado 8 veces en ángulos incrementales de 45°, dando 64 patrones de cada uno de los dígitos usados (1-8), posteriormente se insertó ruido con una probabilidad de 0.01 a 0.05, a bloques de imágenes del mismo tamaño, lo que resultó en un conjunto de 384 patrones. Fue utilizado un algoritmo evolutivo de redes neuronales artificiales para hacer evolucionar la arquitectura de ellas. Se tuvieron 10 individuos por población, evolucionados por 100 generaciones como máximo, donde la evolución puede ser terminada si se llega a un valor de clasificación de cero. Los individuos obtuvieron 144 neuronas de entradas y 8 de salida, dado que las imágenes fueron redimensionadas a matrices binarias de 12x12 bits, con un código binario de 8 bits para cada patrón. Los experimentos fueron repetidos 30 veces por cuestiones estadísticas. El conjunto de prueba de las RNAs consistió en el último bloque de 64 patrones, los cuales presentan la mayor cantidad de ruido (0.05), y el resto fueron usados para diseñar y entrenar las redes neuronales.

5. Resultados

Los resultados obtenidos del experimento realizado muestra que en promedio, las redes neuronales artificiales tardaron 6 generaciones en poder clasificar correctamente todos los patrones del conjunto de prueba. El método usado para determinar el nivel de clasificación fue “Winner takes all”. En las figuras 12–15 se muestra el error, conexiones, mutaciones y nodos ocultos en promedio de las redes neuronales evolucionadas durante 200 generaciones de evolución. Como se puede ver en la figura 12, el error fue disminuyendo conforme avanzaba la evolución. Donde cabe resaltar que el error usado en el NRMSE, Raíz cuadrada del error medio cuadrático normalizado, encargado de determinar el error de adaptabilidad de las redes y como se comentó anteriormente, el error de clasificación fue de cero a partir de la sexta generación.

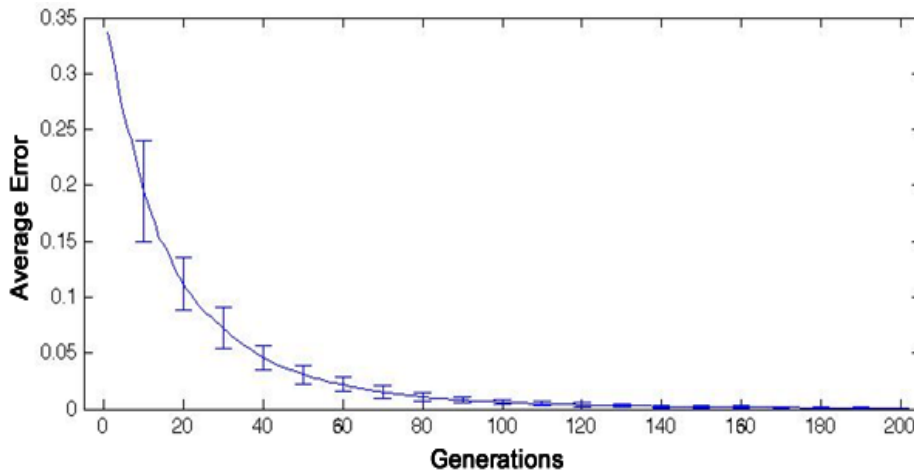


Fig. 12. Error NRMSE promedio durante evolución.

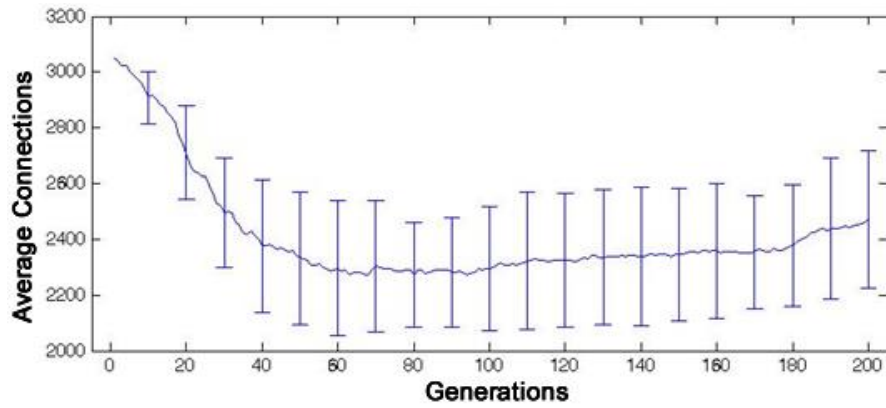


Fig. 13. Número de conexiones en promedio durante evolución.

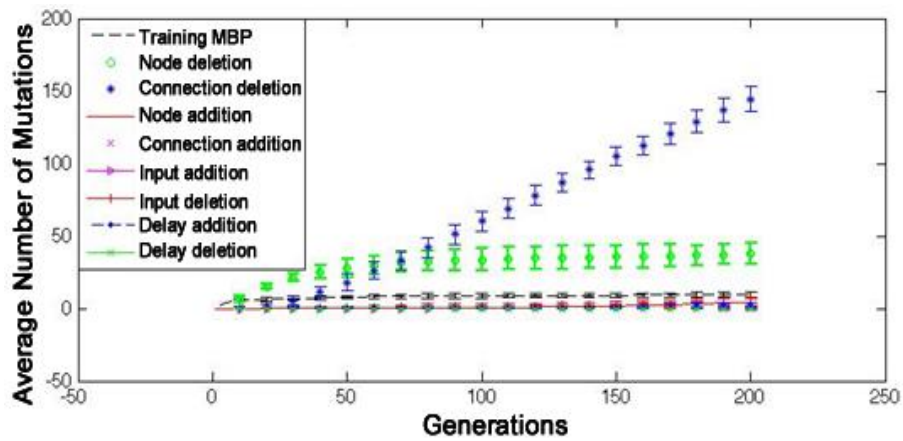


Fig. 14. Número de mutaciones aplicadas a la población en las 200 generaciones de evolución.

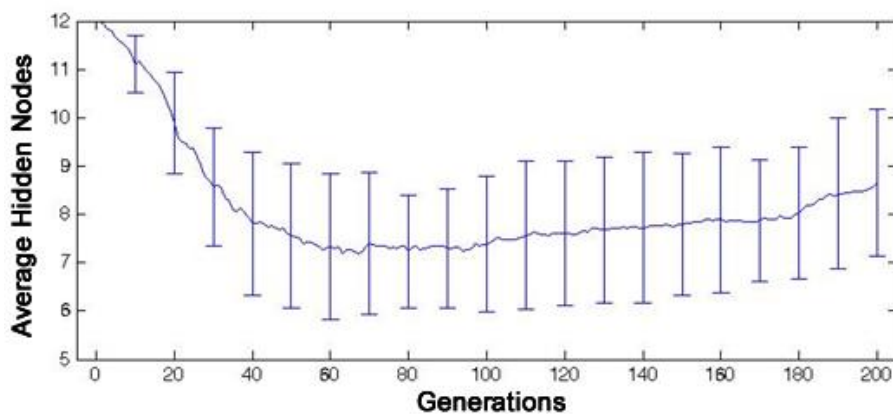


Fig. 15. Número de nodos ocultos en promedio por población por generación.

6. Conclusiones y trabajo a futuro

En este trabajo se presenta una configuración experimental, la cual con ayuda de redes neuronales artificiales fue posible reconocer 8 dígitos los cuales son usados como comandos gráficos para un Drone, donde este puede despegar y desplazarse. Las redes neuronales no mostraron ninguna dificultad para clasificar correctamente los datos, dada la simplicidad de ellos, en comparación con el reconocimiento de caracteres escritos, la cual es una tarea más complicada de resolver. Como trabajo a futuro, se tiene contemplada la aplicación de estos algoritmos para desarrollar el vuelo completamente autónomo del dispositivo y su posterior adaptación y aplicación en diversos entornos. Estos entornos pueden ser ambientes con mucho ruido ya sea en interiores o en exteriores. También se espera presentar más resultados en futuras publicaciones.

Agradecimientos. Se agradece al laboratorio de Cómputo de Alto rendimiento del Centro Universitario UAEM Valle de México por las facilidades prestadas para el uso del Clúster de computadoras para llevar a cabo estos experimentos.

Referencias

1. García Carrillo, L.R.: Modeling the Quad-Rotor Mini-Rotorcraft. *Advances in Industrial Control*, Springer-Verlag London, Vol. 46. pp. 23–35 (2013)
2. Raffo Guilherme, V.: Modelado y Control de un helicóptero Quadrotor. *Universidad de Sevilla Technology*, pp. 35–53 (2007)
3. Peña Giraldo, M.V.: Modelamiento Dinámico y Control LQR de un Quadrotor, *Revista Avances, Investigación en Ingeniería*. Universidad Nacional de Colombia, No. 13, pp. 71–86 (2010)
4. Ramírez Q., J.A., Chacón M., M.I.: Redes Neuronales para el procesamiento de imágenes, una revisión de la última década, *Revista de Ingeniería Eléctrica y Computación*, Vol. 9, No. 1, pp. 7–16 (2011)
5. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press (2005)
6. Coates, A.: Text Detection and Character Recognition in Scene Images with Unsupervised Feature Language, *IEEE Computer Society Washington, DC, USA*, pp. 440–445 (2011)
7. Sánchez Bautista: Diseño y construcción de un Robot Móvil Autónomo contra Incendios. *Universidad Autónoma de Puebla* (2003)
8. Djkshoorn, N.: Simultaneous localization and Mapping with the AR. Drone. Master Thesis for the graduation in Artificial Intelligence, *Universiteit Van Amsterdam* (2012)
9. Castejón Nieto A.: Reconocimiento Visual de Instrucciones de un entorno 3D para un Robot Autónomo. *Universidad Complutense de Madrid Facultad de Informática* (2005)
10. Bielecki, A.: Recognition of two-dimensional representation of urban environment for autonomous flying agents. *Expert Systems with applications*, Elsevier, pp. 3623–3633 (2012)

11. Soto Guerrero, D.: Interacción Hombre-Robot con vehículos aéreos no tripulados basada en visión. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Laboratorio de Tecnologías de la Información. Tesis de Maestría en Ciencias en Computación (2012)
12. Barták, R.: On Autonomous Landing of AR Drone: Hands-on Experience. Charles University in Prague, Faculty of Mathematics and Physics, No. 25 pp. 400–405 (2011)
13. Gordillo Erazo: Sistema de Reconocimiento de Patrones Visuales Basado en Técnicas de Procesamiento de Imágenes y Redes Neuronales, Telematique, Revista Electrónica de Estudios Telemáticos, Vol. 3 (2004)

Red de sensores inalámbrica para la detección de incendios forestales

Jorge Antonio Atempa Camacho, Arnoldo Díaz-Ramírez,
Rafael Iván Ayala Figueroa, Verónica Quintero Rosas

Instituto Tecnológico de Mexicali, Departamento de Sistemas y Computación,
México

{jatempa, adiaz, rafaelivan, veronicaquintero}@itmexicali.edu.mx

Resumen. El problema de los incendios forestales es una amenaza constante para los ecosistemas del planeta. Su incidencia aumenta durante el verano, destruye su alrededor y provoca costos económicos incalculables en la recuperación de zonas boscosas, entre otros. Una red de sensores inalámbrica es una tecnología emergente con la capacidad de extraer datos de las condiciones ambientales del entorno para su procesamiento. La fusión de información permite combinar diferentes fuentes de datos para mejorar la calidad de la respuesta ante un evento de interés. En este artículo se propone un método para la detección de incendios forestales, utilizando una red de sensores inalámbrica y métodos de fusión de información. Su principal contribución es la implementación de un algoritmo de baja complejidad computacional con la capacidad de detectar un evento de incendio utilizando tan solo sensores de temperatura y humedad. La evaluación del método propuesto muestra una tasa de detección de incendios del 100 % cuando los nodos de la red no están expuestos directamente a los rayos del sol.

Palabras clave: Wireless, sensores, redes, inalámbricas, WSN.

Wireless Network of Sensors for Detection of Forest Fires

Abstract. Forest fires represent a constant threaten to the world's ecosystems. They tend to occur more often in the summer season, destroying forest and urban areas, and causing incalculable economic and ecological damages. Wireless sensor networks are an emerging technology that is used to sense the environmental conditions of a predefined area, and to process the gathered data to detect events of interest. In addition, information fusion techniques use the data collected by different sources (i.e., sensors) and combine them to improve the human or automated decision-making process. In this paper, an early forest fire detection method is proposed, which is based on the use of a wireless sensor network and information fusion techniques. The main contribution of the proposed method is that it has a low computational complexity, which is appropriate for implementing real-time detection and notification systems. Also, it is a low-cost method since it uses only two sensors: temperature and relative humidity. The results of the evaluation of the proposed method showed

a detection rate of 100% when the nodes of the network are not directly exposed to sunlight.

Keywords: Wireless, sensor, networks, WSN.

1. Introducción

Año con año México se ve afectado por el problema de los incendios forestales. Éstos representan una de las principales amenazas a los ecosistemas, ocasionando problemas tales como la erosión del suelo, la contaminación en el aire, la desaparición de especies nativas, la pérdida de vidas humanas y cuantiosos costos económicos para la recuperación de zonas boscosas [1]. De acuerdo con la Comisión Nacional Forestal (CONAFOR) de México, en su reporte anual de 2014, el estado de Baja California se ubicó en tercer lugar a nivel nacional al registrar 102 incendios forestales, los cuales dejaron 15,638.95 hectáreas de superficie afectada [2]. Éste es un problema que afecta a todo el planeta y preocupa a la comunidad científica.

Algunos métodos para la detección de incendios incluyen monitorización desde torres de vigilancia y el uso de imágenes satelitales [3, 4]. Desafortunadamente, estos no son eficientes debido a varias razones, tales como los costos elevados en infraestructura (equipo sofisticado), el que requieran un gran número de personal capacitado y que dificultan la monitorización en tiempo real, ya que cuando el fenómeno es detectado, su velocidad de propagación ha producido niveles de daño incontrolables.

Una red de sensores inalámbrica o WSN, por sus siglas en inglés, es un sistema distribuido compuesto por nodos con capacidad de obtener información de las condiciones ambientales y transmitirla de manera inalámbrica a una estación base, para su procesamiento [5]. Por otra parte, las técnicas de fusión de información permiten mejorar la calidad de la respuesta ante un evento de interés, al combinar las diferentes fuentes de datos (sensores, base de datos, etc.) [6]. En este artículo, se propone un sistema de detección de incendios forestales en su etapa inicial, utilizando una red de sensores inalámbrica y métodos de fusión de información. La principal contribución de este trabajo es la implementación de un algoritmo de baja complejidad computacional, con la capacidad de detectar un evento de incendio utilizando solo la información de dos sensores: temperatura y humedad.

El resto del documento se encuentra organizado de la siguiente manera. En la Sección II se revisa el trabajo relacionado. Posteriormente, en la Sección III, se describe el método propuesto para la detección de incendios forestales. La Sección IV discute los resultados obtenidos al evaluar el método propuesto. Por último, en la Sección V, se presentan las conclusiones y el trabajo futuro.

2. Trabajo relacionado

En la literatura se pueden encontrar diferentes propuestas para la detección de incendios forestales utilizando una WSN. En [7], da Penha *et al.* proponen dos algoritmos basados en técnicas de fusión de información para la detección de

incendios utilizando sensores de luz y temperatura. El primero de los algoritmos utiliza el método de umbrales, mientras que el segundo está basado en la teoría de la evidencia de Dempster- Shafer (DST). Las evaluaciones experimentales consistieron en la exposición del nodo sensor ante una antorcha casera utilizada para simular las condiciones de incendio (fuego artificial), a una distancia de 50 a 100 centímetros. El algoritmo basado en DST reportó un número mayor de falsos negativos con respecto al algoritmo basado en umbrales. Sin embargo, los autores resaltan que éste método tiene la ventaja al permitir una extensión rápida de nuevas evidencias (sensores), permitiendo una mayor tasa de detección. En [8], Liu *et al.* proponen un sistema de detección de incendios utilizando una WSN y una red neuronal artificial. El modelo utiliza cuatro tipos de sensores: temperatura, humedad, luz infrarroja y luz visible. Además, cada nodo sensor es alimentado por una celda solar. El objetivo del sistema es poder realizar detecciones tanto en escenarios diurnos como nocturnos. Sin embargo, reportan dificultades para poder distinguir incendios en estos escenarios. Por tal motivo, se implementó una red neuronal artificial con multicapa de retropropagación, para fusionar los datos obtenidos por los nodos sensores. Aunque no se presentan detalles de las evaluaciones experimentales, se mencionan cuatro tipos de escenarios: 1) Nodo separado del fuego a 10 cm. 2) Nodo separado del fuego a 20 cm. 3) Nodo separado del fuego a 30 cm. y 4) 'Normal', que representa a un nodo sin presencia del fuego a su alrededor.

Los resultados manifiestan una alta tasa de detección en escenarios de incendio y cero falsas alarmas en escenarios normales. En [9] A. Ko *et al.* describen un sistema de detección de incendios forestales inteligente. El sistema cuenta con tres subsistemas: cámaras, red de sensores inalámbrica y subsistemas móviles. Estos subsistemas fueron diseñados para verificar los datos obtenidos de forma independiente, y así aumentar la fiabilidad y reducir al mínimo la tasa de falsas alarmas. El subsistema de la cámara combina y analiza imágenes de alta resolución, obtenidas por cámaras infrarrojas de onda corta y cámaras térmicas, para detectar el fuego en el bosque desde varias ubicaciones. La WSN utiliza nodos colocados en lugares estratégicos en todo el bosque, para obtener información sobre la temperatura, humedad, velocidad del viento y otra información local. El subsistema móvil con capacidad de maniobra ofrece datos verificables, que no se pueden obtener a través de los subsistemas estáticos. Sin embargo, el autor no presenta resultados de las evaluaciones.

3. Modelo de detección propuesto

Como parte del desarrollo del sistema de detección de incendios, el primer paso fue conocer las características del entorno en condiciones normales (sin presencia de incendio) y en condiciones de incendio. Con el objetivo de desarrollar una propuesta robusta, durante el verano (etapa del año más propensa a los incendios forestales) se colectó información de la temperatura y la humedad relativa en condiciones normales, en una zona arbolada del campus universitario. El análisis de la información arrojó las siguientes observaciones:

1. En condiciones normales, los valores de temperatura y humedad manifiestan un comportamiento cíclico durante las diferentes etapas del día.

2. En condiciones normales y condiciones de incendio, los valores de temperatura y humedad mantienen una relación inversamente proporcional entre sus magnitudes; es decir, cuando la temperatura se incrementa, la humedad disminuye y viceversa.

3. En condiciones normales, los valores de temperatura y humedad varían muy lentamente durante las diferentes etapas del día. Por otro lado, en condiciones de incendio, la tasa de variación es mayor y se manifiesta en mucho menos tiempo. En la Fig. 1, se observan las mediciones obtenidas de temperatura y humedad en condiciones normales (T normal y H normal), así como en condiciones de incendio (T fuego y H fuego), durante el mismo intervalo de tiempo (seis minutos).

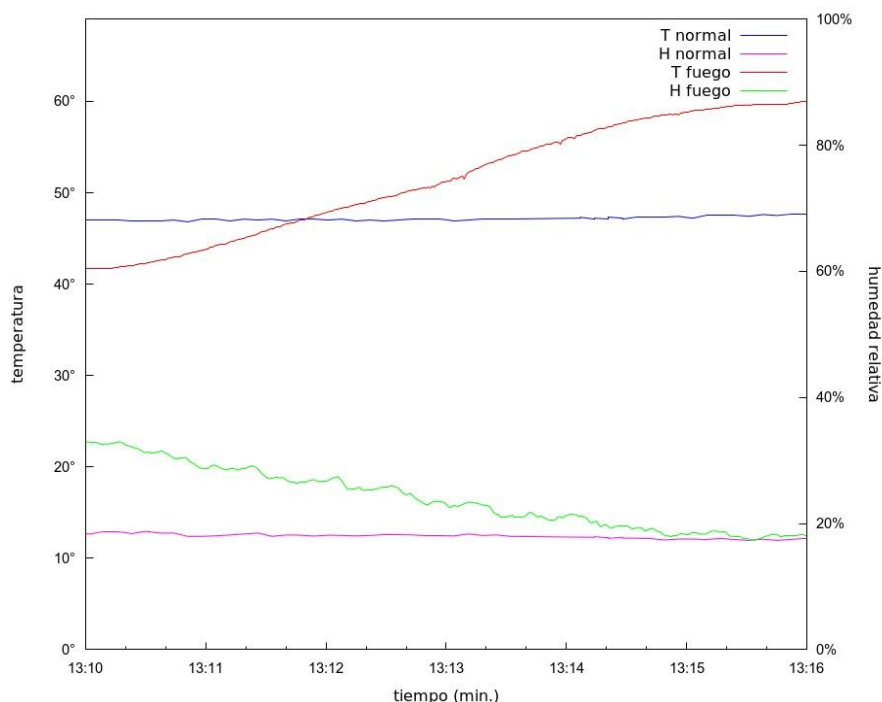


Fig. 1. Valores de temperatura y humedad en condiciones normales e incendio.

4. Cuando un nodo sensor es expuesto a los efectos de los rayos del sol, la temperatura muestra una razón de cambio similar a la producida por un incendio, como se muestra en la Fig. 2.

Tomando en cuenta estas observaciones, se investigaron diferentes técnicas para poder caracterizar el comportamiento de un incendio forestal. En una propuesta inicial, se construyó una función que relacionara el comportamiento de la temperatura y la humedad en un incendio, a la que se nombró función base. Posteriormente, esta función se utilizó para comparar los valores obtenidos por los sensores cuando se registraba un evento que pudiera ser un incendio. A través del uso de técnicas de interpolación y del uso de la teoría de la evidencia de Dempster-Shafer, se determinaba si el evento se trataba de un incendio o no [10]. Sin embargo, a pesar de que se obtuvo una buena tasa de detección, el método no era robusto si las

condiciones ambientales variaban mucho con respecto a las utilizadas para crear la función base.

En este trabajo, se propone la construcción de la función base utilizando el análisis de regresión. El análisis de regresión ofrece entre sus ventajas la representación de los datos a través del ajuste de una función matemática (línea recta, una función polinomial, una función exponencial, etc.), además de que facilita la estimación de valores desconocidos. En este trabajo, se utilizó el análisis de regresión simple, en el cual existen una variable dependiente y una variable independiente.

Tomando en consideración que los valores de temperatura y humedad relativa mantienen una relación inversamente proporcional en un incendio, se determinó construir dos funciones base, que representen su comportamiento en el incendio, de forma independiente. Del análisis de los parámetros bajo estudio, se encontró que la característica más relevante del incendio es la razón de cambio de los valores de temperatura y humedad con respecto al tiempo. Por tal motivo, al utilizar el análisis de regresión para construir las funciones base, se considera al tiempo como la variable independiente, y a la temperatura y humedad como variables dependientes. Estas funciones son llamadas a partir de ahora como $T(t)$ y $H(t)$, respectivamente, y constituyen el modelo base.

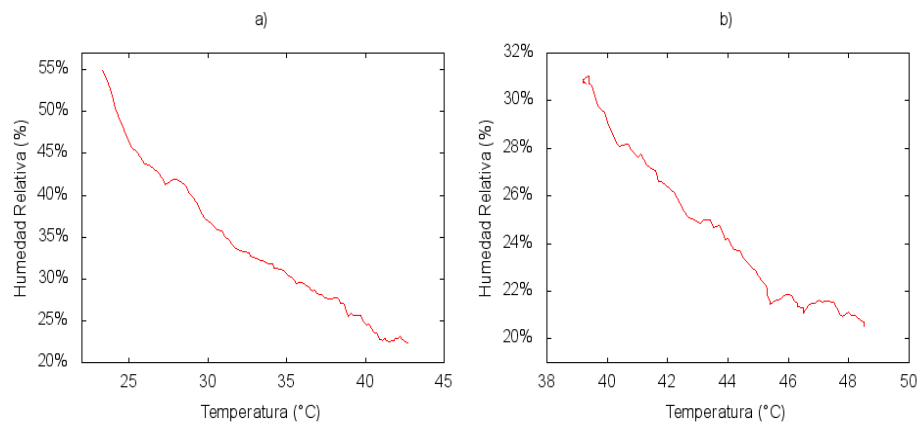


Fig. 2. a) Nodo expuesto a los rayos del sol, b) Nodo expuesto a fuego artificial.

Para la construcción del modelo base, se propone que se utilicen mediciones de temperatura y humedad en experimentos de incendios, considerando alguna época del año y una franja horaria específica. Por ejemplo, pueden construirse modelos base para la temporada de verano, y en diferentes etapas del día (mañana y tarde), que de acuerdo al análisis de datos históricos, fuesen los que han mostrado mayor incidencia de eventos. En la siguiente sección se mostrará un ejemplo de la construcción del modelo base.

El análisis de regresión utiliza el método de mínimos cuadrados para el ajuste de una función a un conjunto de datos. Este método es una manera estandarizada de medir la variación entre los datos y proporciona una función que minimiza la suma de los cuadrados de las diferencias entre la estimación de un modelo de función ideal (y_e) y los datos obtenidos por una fuente (y_i), como se observa en la Ecuación 14.

$$S = \sum_{i=1}^n (y_e - y_i)^2 \quad (1)$$

Por ejemplo, utilizando los datos experimentales que se muestran en la Fig. 3, podemos percibir en los mismos un comportamiento ascendente. El método de mínimos cuadrados permite determinar una función que se ajuste y represente el comportamiento de los datos, como se muestra en la Fig. 3.

La primer opción de función ideal para ajustarse a estos datos es una línea recta; es decir, una ecuación de la forma $y_e = mx + b$. Al sustituir este valor en la Ecuación 1, se obtiene la Ecuación 2.

$$S = \sum_{i=1}^n (mx_i + b - y_i)^2 \quad (2)$$

Esta nueva ecuación proporciona dos incógnitas: la pendiente (m) y la constante (b). Derivando con respecto a cada una de éstas, se obtienen las Ecuaciones 3 y 4.

$$\frac{\delta S}{\delta m} = m \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i - \sum_{i=1}^n x_i y_i \quad (3)$$

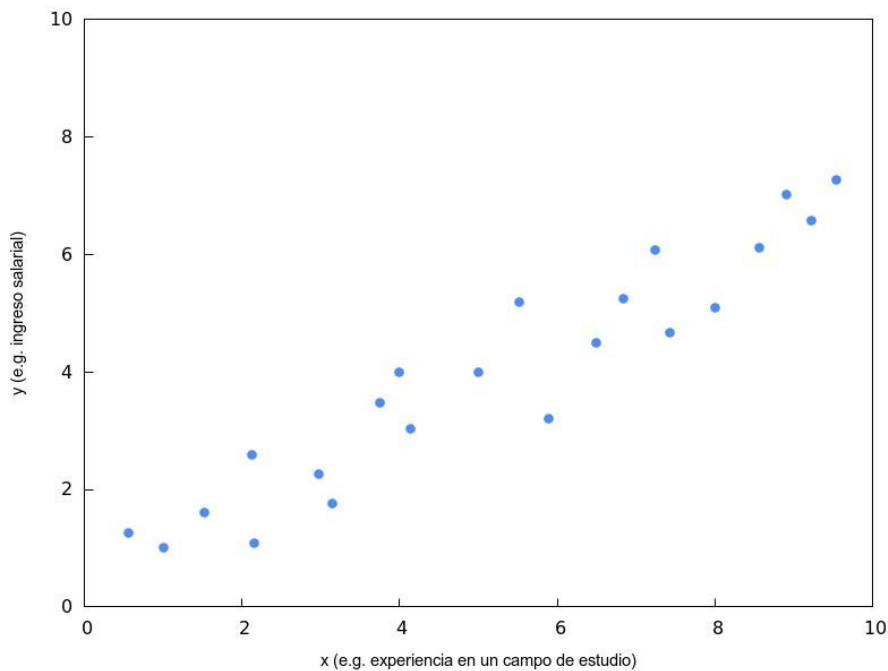


Fig. 3. Ejemplo de datos experimentales.

$$\frac{\delta S}{\delta b} = m \sum_{i=1}^n x_i + bn \sum_{i=1}^n y_i \quad (4)$$

Al resolver este sistema de ecuaciones por el método de determinantes, obtenemos las Ecuaciones 5 y 6:

$$m = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (5)$$

$$b = \frac{\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i}{n} \quad (6)$$

El último paso, consiste en desarrollar y sustituir los resultados de las Ecuaciones 5 y 6 en la Ecuación 2. Esto proporciona una función cuya estimación a los valores dados es lo más cercana posible, como se muestra en la Fig. 4.

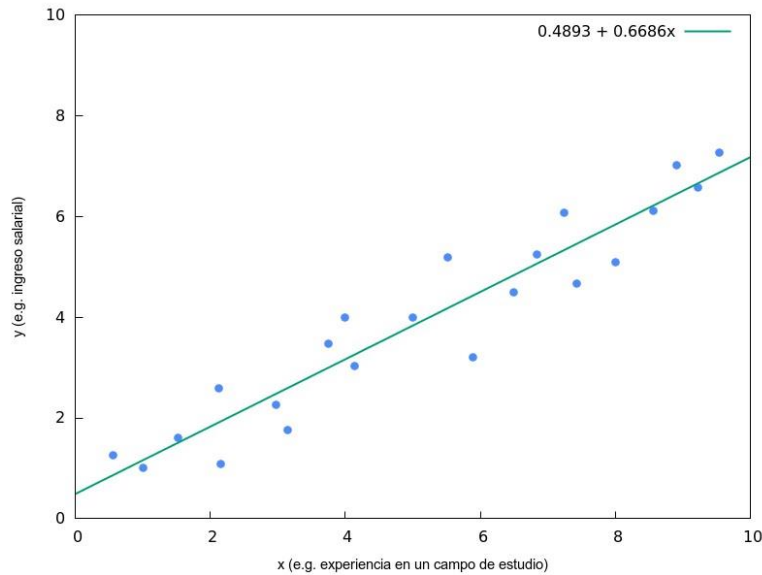


Fig. 4. Función obtenida por el método de mínimos cuadrados.

Sin embargo, la relación entre las variables no siempre se manifiesta de forma lineal como en el ejemplo anterior. Para estos casos existen alternativas como funciones de segundo orden o parábolas, funciones de tercer orden o cúbicas, etcétera. Una manera más general de representar funciones no lineales de orden superior se muestra en la Ecuación 7:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n. \quad (7)$$

Cabe destacar que también es posible ajustar funciones exponenciales, logarítmicas, entre otras. Por otro lado, uno de los parámetros que validan el resultado del análisis de regresión es la bondad de ajuste. Si los datos se encuentran muy dispersos, la bondad de ajuste será baja; en caso contrario si la dispersión es pequeña entonces la bondad de ajuste será alta. A éste valor se le llama coeficiente de

determinación o r^2 . El coeficiente de determinación es un valor que oscila entre 0 y 1, y se calcula de la siguiente manera:

$$r^2 = 1 - \frac{\text{Suma de cuadrados de los residuos}}{\text{Suma de cuadrado total}} \quad (8)$$

El coeficiente de determinación es un índice que determina la calidad del ajuste de la función obtenida por el análisis de regresión, con respecto una serie de datos en particular. Cuanto más cercano sea r^2 a 1, mejor será su calidad de ajuste.

El objetivo del modelo base es contar con una referencia que proporcione mayor precisión al momento de representar las condiciones ambientales existentes en un incendio. Esto permite la posibilidad de no depender de magnitudes fijas o umbrales, sino más bien de la razón de cambio que se produce cuando el fenómeno de incendio afecta al entorno. En otras palabras, de acuerdo con nuestras observaciones durante un incendio forestal, los valores de temperatura y humedad se comportan siempre de la misma manera: ascendente para el caso de la temperatura y descendente para el caso de la humedad relativa, con una razón de cambio o pendiente similar, como puede observarse en la Fig. 5. Contar con un modelo que represente éste comportamiento (independientemente de las magnitudes), permite comparar los datos recibidos por los nodos sensores para afirmar o descartar sí la variación observada corresponde a un incendio. Para tal efecto se propone la siguiente arquitectura, la cual, está compuesta por tres módulos: red de sensores inalámbrica, middleware y sistema de detección de incendios.

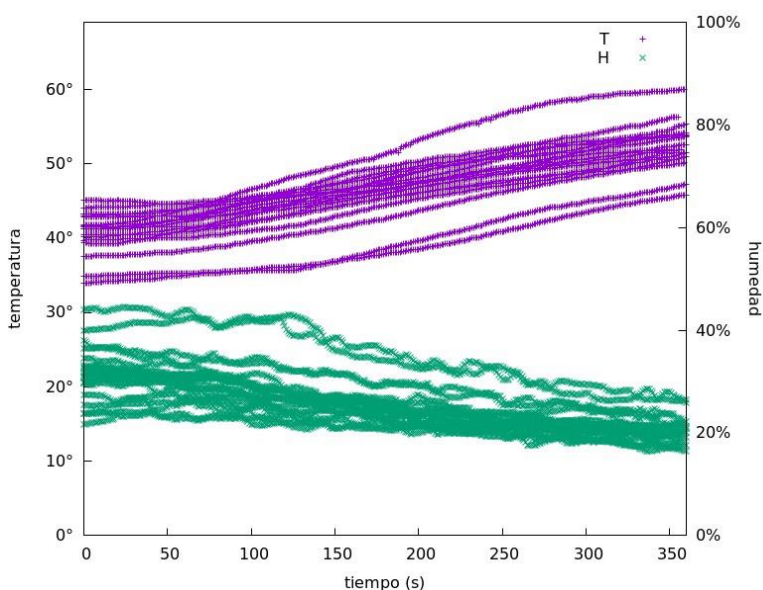


Fig. 5. Valores de temperatura y humedad en un incendio.

En términos generales, el modelo propuesto funciona de la siguiente manera: 1) Los nodos sensores recaban mediciones del entorno periódicamente; 2) cuando detectan un incremento en la temperatura, transmiten la información a la estación

base; 3) la estación base reenvía los datos al servidor para su almacenamiento en una base de datos; 4) el servidor toma la información más reciente de la base de datos y realiza una comparación entre los datos con respecto a un modelo base previamente definido; y 5) de acuerdo al grado de similitud entre los datos con respecto al modelo base, el sistema determina si existe o no un incendio en la zona monitorizada. A continuación se describen cada una de las etapas de manera más detallada.

En la red de sensores inalámbrica, cada nodo realiza una lectura de la temperatura utilizando un período de muestreo constante P . Cada lectura es almacenada en una estructura FIFO (*First-In, First-Out*) llamada W_T que utiliza una ventana deslizante de tamaño n . Cuando W_T se llena, se verifica si la lectura más reciente T_{n+1} ha manifestado un cambio (incremento o decremento), a través de la siguiente razón representada por Ω :

$$\Omega = \frac{T_{n+1}}{\mu(W_T)}, \quad (9)$$

donde por otro lado $\mu(W_T)$ equivale al promedio del conjunto de datos sobre la temperatura almacenado en W_T . El resultado de Ω , será comparado con un umbral de temperatura denominado como $T_{threshold}$. Sí y solo sí, Ω es mayor a $T_{threshold}$, el nodo sensor realiza una lectura de la humedad H y posteriormente transmite un paquete de datos a la estación base con la siguiente información: identificador del nodo id , muestra de temperatura T , muestra de humedad H y una estampa de tiempo t para identificar el momento en el que se generó el evento. Para el primer paquete transmitido, el valor de t será igual a cero, y para casos subsecuentes el valor de t estará en función de P . En caso no superar el umbral de temperatura, el nodo sensor debe continuar leyendo y evaluando muestras. Éste procedimiento se describe en el Algoritmo 1.

Algoritmo 1 Pre-procesamiento en el nodo sensor

Sea T = temperatura, P = período de muestreo, W_T = buffer de muestras de temperatura, n = tamaño de la ventana deslizante, $\Omega = 0$, $T_{threshold}$ = umbral de temperatura, id = identificador del nodo sensor, H = humedad, $t = 0$, $i = 1$;

1. while (1) {
2. T = Leer la temperatura ambiental;
3. if ($i \leq n$) {
4. $W_T[i] = T$;
5. $i = i + 1$;
6. } else {
7. $\Omega = \frac{T}{\mu(W_T)}$
8. while ($\Omega > T_{threshold}$) {
9. H = Leer la humedad ambiental;
10. Enviar(id, T, H, t);
11. Remover T más antiguo de W_T ;
12. Almacenar T más reciente en W_T ;
13. $t = t + P$;
14. } T = Leer la temperatura ambiental;

```

15.       $\Omega = \frac{T}{\mu(W_T)}$ 
16.      } //fin del ciclo while
17.       $t = 0;$ 
18.       $i = 1;$ 
19.      } //fin else
20.      } // Esperar el siguiente período
    
```

Cuando el nodo sensor transmite un paquete a la estación base, la información se almacena en una base de datos para su inmediato análisis. La función del middleware es establecer mecanismos de comunicación entre la estación base y el servidor. El middleware asigna un tiempo de arribo compuesto por la fecha y hora del servidor para cada paquete recibido de acuerdo con el identificador del nodo y el equivalente a su respectivo valor t .

Posteriormente, el servidor obtiene la información más reciente de la base de datos, tomando en cuenta que cada paquete obtenido por los nodos sensores se compone de los siguientes datos: id , T , H y t . Para describir el flujo del sistema de detección de incendios se utiliza A como el conjunto de información menos reciente o información que rebasó un umbral de temperatura por primera vez y como B la información más reciente o información subsecuente acerca de las condiciones del entorno.

El primer paso es verificar que el tiempo de análisis transcurrido Y , sea menor al tiempo de análisis límite Θ , esto con el objetivo de analizar durante un lapso de tiempo balanceado que permita contar con la información necesaria para tomar una decisión. Al inicio del análisis $Y = 0$, para actualizar su valor debemos acumular el resultado de las diferencias entre $t_B - t_A$, como se muestra en la Ecuación 10:

$$Y = Y + (t_B - t_A). \quad (10)$$

Suponiendo que una vez recibido un par de muestras (A y B), a su diferencia en el tiempo se le denomina K , por lo que $Y = K$. Para cada nueva muestra se repite el procedimiento mientras el valor de Y sea menor que Θ (valor definido por el usuario).

Una vez calculado Y y validado que es menor a Θ , el siguiente paso es calcular la razón de cambio de los valores de temperatura y humedad de A y B utilizando la fórmula de la pendiente, es decir, m_T y m_H como se muestra en las Ecuaciones 11 y 12.

$$m_T = \frac{T_B - T_A}{t_B - t_A}, \quad (11)$$

$$m_H = \frac{H_B - H_A}{t_B - t_A}. \quad (12)$$

Después debe calcularse la derivada de las funciones que representan el modelo base, es decir, $T(t)$ y $H(t)$. Sin embargo, cabe mencionar que $T'(t)$ y $H'(t)$ deben obtener un resultado equivalente al valor de Y , por lo cual, éstas se sustituyen por $T'(Y)$ y $H'(Y)$ Posteriormente se utiliza la técnica del error cuadrático promedio utilizando un número de evaluaciones límite denominado como $emax$. El error cuadrático promedio es utilizado para medir el error entre un estimador (funciones del

modelo base) y una estimación (datos provenientes del nodo sensor). Este valor se calcula como se muestra en las Ecuaciones 13 y 14:

$$MSE_T = \frac{\sum_{i=0}^{emax} (T'_i(\gamma) - m_{T_i})^2}{emax}, \quad (13)$$

$$MSE_H = \frac{\sum_{i=0}^{emax} (H'_i(\gamma) - m_{H_i})^2}{emax}. \quad (14)$$

Con el objetivo de clasificar los resultados del error cuadrático promedio, se definió una tabla que proporciona un valor de masa con dominio entre 0 y 1, donde cero representa un alto grado de similitud (entre 90 y 100 por ciento), mientras que 1 representa un grado casi nulo de similitud con respecto a los datos evaluados. El valor de masa es utilizado como resumen de la serie de comparaciones del MSE. El último paso, consiste en obtener el promedio de las masas $\frac{(mass_T + mass_H)}{2}$ y compararlo con una constante α . Este último valor es utilizado para determinar si el sistema debe o no, enviar una alerta de incendio al usuario.

4. Evaluación

Con la finalidad de evaluar el desempeño del modelo propuesto, se llevaron a cabo una serie de experimentos simulando incendios forestales. En esta sección se presentan los resultados obtenidos.

Para llevar a cabo el conjunto de experimentos de incendios fueron utilizados nodos sensores con plataforma de hardware IRIS, con una placa sensora modelo MTS420/400CC [11]. Esta placa tiene integrados cinco tipos de sensores ambientales: humedad relativa, temperatura, presión barométrica, intensidad luminosa y acelerómetro. En cuestiones de software, en el servidor se utilizó Ubuntu Linux versión 12.04. El middleware fue desarrollado en Java y la base de datos en MySQL. Por último, para la WSN se utilizó el sistema operativo TinyOS en su versión 2.1.2.

La metodología utilizada consistió en colocar el nodo sensor sobre la parte media del tronco de un árbol, a una altura aproximadamente de tres metros. Los experimentos se llevaron a cabo durante los meses de Abril y Mayo de 2015. Por otra parte, la recolección de datos fue dividida en dos etapas: 1) Creación de modelo base y 2) Experimentos de fuego controlado. A continuación se describe cada una de ellas.

Tabla 1. Parámetros utilizados en experimentos

Parámetro	Descripción	Valor
n	Tamaño de la ventana W_T	15
$T_{\text{threshold}}$	Umbral de temperatura	1.01
P	Período de muestreo del nodo sensor	6 segundos
Θ	Tiempo de análisis límite	360 segundos
emax	No. de evaluaciones máximo para el MSE	5
α	Umbral utilizado para determinar un incendio	0.7

1. Modelo base: Con el objetivo de contar con suficiente información para el desarrollo de las funciones base $T(t)$ y $H(t)$, en esta etapa el nodo sensor fue

expuesto a condiciones de fuego artificial utilizando una antorcha para jardín. La duración del experimento fue de 6 minutos utilizando un período de muestreo de 1 segundo. En esta etapa se realizaron 20 experimentos.

2. Experimentos de fuego controlado: Se llevaron a cabo series de experimentos con diferentes períodos de muestreo: 3, 4, 6 y 8 segundos. El objetivo fue investigar el período de muestreo con mejor desempeño para la detección del evento. En esta etapa se realizaron 32 experimentos (8 por cada tipo de período de muestreo utilizado). La finalidad fue determinar el período de muestreo a utilizar en la siguiente etapa de experimentos.

Para construir el nuevo modelo base, se implementó el ajuste de curva por análisis de regresión simple con la ayuda del software de análisis estadístico SPSS de IBM [12]. Con los datos recabados, se obtuvo el siguiente modelo base:

$$T(t) = -2.477 \times 10^{-7} t^3 + 1.12 \times 10^{-4} t^2 + 2.2441 \times 10^{-2} t + 28.8861, \quad (15)$$

$$H(t) = 2.26 \times 10^{-8} t^3 - 1.7 \times 10^{-5} t^2 - 2.7444 \times 10^{-2} t + 17.505. \quad (16)$$

Cabe mencionar que los coeficientes de determinación fueron de 0.87 para la función $T(t)$ y 0.68 para la función $H(t)$.

Posteriormente, se llevaron a cabo 50 experimentos de incendio, utilizando los parámetros que se muestran en la Tabla 1, 30 de ellos escenarios de sombra y con nodo sensor expuesto a fuego artificial. Los 20 restantes fueron eventos sin incendio y con el nodo sensor fue expuesto a los rayos del sol, sin protección alguna. Estos valores fueron procesados utilizando el modelo base descrito por las Ecuaciones 15 y 16. El modelo obtuvo una tasa de detección del 100% en escenarios de sombra. Al realizar el análisis de los datos obtenidos por un nodo sensor expuesto a los rayos del sol, se obtuvo una tasa de 100% de falsos positivos. Sin embargo para evitar esto, se puede cubrir al nodo sensor con protección especial para evitar la exposición directa a la luz solar. Para comprobarlo, se llevaron a cabo un nuevo conjunto de 50 experimentos con el nodo sensor expuesto a los rayos del sol, pero cubriendo a éste con una protección. Al utilizar el método propuesto en estas condiciones, se obtuvo una tasa de detección del 100%.

5. Conclusiones y trabajo futuro

En este trabajo se propuso un sistema de detección de incendios forestales en su etapa inicial utilizando una red de sensores inalámbrica y métodos de fusión de información. Su principal característica es que utiliza tan solo información de los sensores de temperatura y humedad. Su baja complejidad computacional lo hace ideal para monitorizar zonas de riesgo y emitir alarmas en tiempo real. El modelo requiere del desarrollo modelo base que represente las características de las condiciones ambientales existentes en un incendio. El modelo base se construye utilizando análisis de regresión. Cuando el sistema detecta valores de temperatura y humedad que pudieran representar un incendio, compara los valores recabados con el modelo base para determinar si existe o no un incendio. Los resultados de la evaluación del modelo mostraron una tasa de detección del 100% cuando los nodos no están expuestos directamente a los rayos del sol. Como trabajo futuro, se busca extender el modelo

considerando el consumo de energía de la WSN, a través de la distribución de los nodos en clusters y el uso de sensado distribuido.

Referencias

1. Ali, A.A., Carcaillet, C., Gauthier, S., Hély, C., Le Goff, H., Terrier, A., Bergeron Yves, G., Martin, P.: Fire in managed forests of eastern Canada: Risks and options. *Forest Ecology and Management*, 294, pp. 238–249 (2013)
2. CONAFOR. Incendios forestales en México temporada, <http://www.conafor.gob.mx/web/temas-forestales/incendios/> (2014)
3. AVHRR. <http://noaasis.noaa.gov/NOAASIS> (2015)
4. MODIS. <http://modis.gsfc.nasa.gov/> (2015)
5. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey, *Computer Networks*, 52(12), pp. 2292–2330 (2008)
6. Nakamura, E.F., Loureiro, A.F., Frery, A.C.: Information fusion for wireless sensor networks: Methods, models, and classification, *ACM Comput. Surv.*, 39(3) (September 2007)
7. da Penha O.S., Nakamura, E.F.: Fusing light and temperature data for fire detection. In: *Computers and Communications (ISCC), IEEE Symposium on*, pp. 107–112 (June 2010)
8. Liu, Y., Gu, Y., Chen, G., Ji, Y., Li, J.: A novel accurate forest fire detection system using wireless sensor networks. In: *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*, pp.52–59 (Dec 2011)
9. Ko, A., Lee, N.M.Y., Sham, R.P.S., So, C.M., Kwok, S.C.F.: Intelligent wireless sensor Network for wildfire detection. 158, pp. 137–148 (2012)
10. Díaz-Ramírez, A., Tafoya, L.A., Atempa, J.A., Mejía-Alvarez, P.: Wireless sensor networks and fusion information methods for forest fire detection, *Procedia Technology*, 3(0), pp.69–79. In: *The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science* (2012)
11. MEMSIC. Product datasheet. <http://www.memsic.com/> (Enero 2014)
12. IBM. <http://www-01.ibm.com/software/analytics/spss/> (2014)

Simulación 3D Basada en Agentes de Robo y Asalto a Transeúntes

Fabián Beltrán Osnaya, Héctor Rafael Orozco Aguirre,
Victor Manuel Landassuri Moreno

Universidad Autónoma del Estado de México, Centro Universitario UAEM Valle de México,
Atizapán de Zaragoza, Estado de México, México

fabian.b.osnaya@gmail.com, hrorozcoa@uaemex.mx, vmlandassurim@uaemex.mx

Resumen. Los efectos de la delincuencia son diversos y complejos, desde los traumas psicológicos y físicos que enfrentan las víctimas, a los impactos negativos en la economía e imagen externa de un país. Desde una perspectiva criminológica, la perpetración de un crimen es causada por un conjunto de factores tales como las circunstancias situacionales, contexto social, y los factores a nivel individual. En este artículo, se presenta un nuevo modelo de simulación 3D en tiempo real de un vecindario apoyado en la criminología científica, para apreciar cómo agentes policías, víctimas y delincuentes, se ven inmersos en situaciones de robo y asalto en la vía pública, así como de inspección y arresto.

Palabras clave: Agentes, robo, asalto, transeúnte, vía pública, AnyLogic.

3D Simulation based on Robbery Agents and Assaults of Passersby

Abstract. The effects of crime are diverse and complex, ranging from psychological and physical trauma faced by victims to the negative impacts on the economy and external image of a country. From a criminological perspective, the commission of a crime is caused by a combination of factors such as situational circumstances, social and individual-level factors. In this paper, a new 3D real-time simulation model for a neighborhood is presented, which is supported by scientific criminology, in order to appreciate how police officers, victims and offenders are involved in situations of robbery and assault on public roads as well as inspection and arrest.

Keywords: Agents, robbery, assault, passersby, AnyLogic.

1. Introducción

En todo el mundo, la acción de caminar o transportarse en la calle o vía pública, implica correr el riesgo de ser asaltado, sobre todo si se vive en las grandes ciudades.

Las urbes de América Latina son un claro ejemplo de ello. En México, el robo es uno de los principales delitos, el cual es derivado de problemas sociales que siguen a la alza. Según las cifras del secretariado ejecutivo [1], del mes de enero al mes de agosto de 2015 se registraron 998,976 presuntos delitos, de los cuales el Estado de México registra 136,621; el Distrito Federal 111,285 y Baja California Norte 69,614; siendo estas tres entidades federativas las que presentan el mayor número de incidencias reportadas [2].

De acuerdo a la procuraduría general de justicia del estado y a las denuncias presentadas ante agencias del ministerio público, la cifra total de robos registrados es de 372,736 lo que equivale al 37.31% de los presuntos delitos reportados [3]. De igual manera, se tiene un registro de robo común con una cifra de 365,044 siendo equivalente al 97.93% con respecto al total de robos registrados. De dicho tipo de robo, resaltan dos de sus categorías: con violencia y sin violencia, ambas a transeúntes, la suma de estas es de 46,296, lo cual es equivalente al 12.68% del robo común registrado. De los robos a transeúntes, 11,888 son sin violencia y 34,408 con violencia; representando el 25.68% y el 74.32% de la suma total de robos, respectivamente [3].

Las cifras obtenidas y mencionadas anteriormente, no indican cuáles son los factores presentes en un robo o asalto en la vía pública, siendo necesario conducir y reforzar investigaciones para obtener orientaciones que permitan emitir probabilidades, causas y efectos de estos problemas sociales.

Este trabajo, está enfocado al área de la criminología científica, mediante él se propone y aplica un nuevo modelo de simulación 3D basada en agentes que pretende dar la pauta de algunas de las causas y probabilidades que originan situaciones de robo y asalto en la vía pública, mediante la definición y generación de comportamientos dinámicos y predefinidos a partir de características dadas a los agentes. En este modelo, se contemplan diversos tipos de robo y asalto a transeúntes. Los robos modelados son el de cartera de hombres y el de bolso de mujeres. En el caso de los asaltos, se tienen de menor a mayor grado de violencia e impacto para la víctima, el asalto simple, el intimidatorio y el agravado.

2. Estado del arte

Los criminólogos han demostrado que las características naturales y el diseño de un entorno tienen un impacto significativo en las comunidades y el crimen que en ellas se suscita [4]. La mayoría de los criminales sólo delinquen si están seguros de lograr su cometido, al estar familiarizados con la zona seleccionada y contando con que otros individuos no intervendrán [5]. La situación socioeconómica de un área es particularmente importante, así los robos y asaltos se producen de manera desproporcionada en las zonas con alto nivel socioeconómico, en especial, cuando estas están cerca de áreas con altas tasas de delincuencia [6], los de tipo violento a menudo se asocian con zonas de alto desempleo y un bajo nivel socioeconómico.

En un intento de explicar lo que impulsa a los delincuentes y el efecto que se tiene sobre las víctimas, se han desarrollado diversas teorías criminológicas, entre ellas la

teoría de las actividades de rutina [7], la cual estipula que por cada crimen que se produce debe estar presente un delincuente motivado y una víctima, además de la ausencia de un guardia capaz de impedirlo. Aunque dicha teoría es de ayuda para establecer lo que los delincuentes, las víctimas y los guardianes hacen en su vida cotidiana, no establece las decisiones de las personas o las acciones que realizan, sin embargo estos factores pueden ser modelados mediante lo que dicen otras teorías.

La teoría del patrón del crimen [8] examina los espacios de actividad que determinarán las áreas que los delincuentes conocen bien para perpetuar un crimen. La teoría de perspectiva de la elección racional [9] sugiere que la decisión de un delincuente puede ser modelada como un proceso formal por el cual los beneficios potenciales de un delito exitoso se comparan contra las pérdidas potenciales si se es aprehendido.

Las ocurrencias del crimen han sido modeladas mediante la técnica estadística de regresión logística, la cual a pesar de haber dado resultados interesantes trabaja en grandes escalas por lo que no toma en cuenta factores locales como: las personas, calles o el entorno de un vecindario. Para hacer frente a este problema se están utilizando simulaciones por computadora las cuales pueden trabajar a escalas mucho más pequeñas, lo cual incluye modelos de interacción espacial y técnicas de micro simulación [10]. Sin embargo, estas técnicas se enfrentan a dificultades con respecto a la incorporación de factores de comportamiento humano. El modelado basado en agentes ofrece la mejor alternativa, ya que estos pueden interactuar entre sí y con su entorno [11]. De esta manera, un agente evalúa individualmente su situación y con base en reglas, toma una decisión sobre qué acción tomar, permitiendo que en él se puedan implementar comportamientos humanos. La tabla siguiente, da un comparativo con los trabajos más similares al aquí presentado, destacando los aportes de éste.

Tabla 1. Comparación con otros modelos de simulación basados en agentes.

Modelo	Escenario	Agentes	Comportamiento	Tiempo real	Plataforma utilizada	Delitos
Canessa y Quezada [12]	Simple	Ladrón Policía	Predefinido	No	Netlogo	Robo simple
Devia [13]	Simple	Ladrón Policía Ciudadano	Semi-dinámico	No	Repast Symphony	Robo simple por sorpresa y con violencia
Amrutha [14]	Muy básico	Ladrón Ciudadano Policía	Predefinido	No	Netlogo	Robo simple en casa
Malleston [15]	Simple	No especificado	Predefinido	No	Netlogo (aún no implementada)	Robo simple
El aquí propuesto	Complejo	Ladrón Policía Ciudadano	Predefinido y dinámico	Sí	AnyLogic	- Asalto simple, intimidatorio y agravado - Robo de cartera y bolso

3. Modelo de simulación 3D propuesto

La Fig. 1 muestra la implementación basada en agentes del modelo propuesto, en la cual se contempla un entorno virtual 3D que fue modelado y programado en la versión para investigador de la plataforma AnyLogic [16]. Este entorno incluye la representación de un escenario para un vecindario ordinario como cualquier otro.

La plataforma AnyLogic fue elegida debido a que es una herramienta idónea para simulaciones 3D basadas en agentes, ya que permite crear y programar comportamientos en escenarios multiagente, para simular fenómenos y problemas sociales, tal es el caso del robo y asalto a transeúntes, mediante el modelado de sistemas complejos que evolucionan con el tiempo. En estos sistemas, el programador da un comportamiento dinámico mediante instrucciones en diagramas de estado a agentes que trabajan de manera independiente o colectiva. Cabe destacar que esta herramienta permite definir diferentes tipos de agentes a partir de sus características, denotadas por parámetros y variables; así como de sus comportamientos, denotados como funciones, que pueden tener cada uno de ellos. El período de simulación de un entorno virtual en AnyLogic es en tiempo real y se mide en segundos, el cual puede ser acelerado para obtener resultados de manera más rápida, siempre en concordancia con un tiempo equivalente del mundo físico.

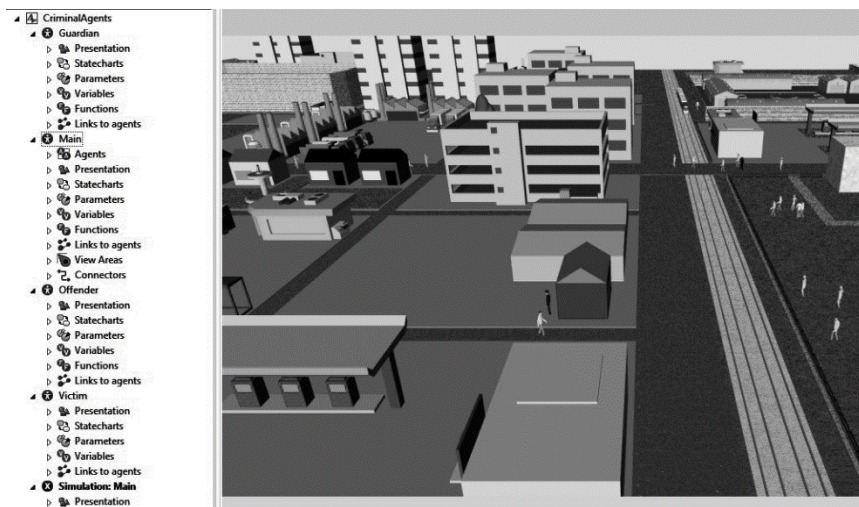


Fig. 1. Creación del modelo de simulación 3D basada en agentes de robo y asalto a transeúntes.

En la vía pública del escenario creado se localizan tres diferentes tipos de agentes, los cuales son descritos como sigue:

- Agentes delincuentes o criminales: quienes siempre están en busca de objetivos (víctimas) que podrían ser fácilmente atacados (ver Fig. 2 y Tabla 2).
- Agentes objetivos o víctimas: cada uno de estos puede ser cualquier agente de la población no delincuyente que se encuentra dentro del radio de acción de un agente

delincuente, con una cierta probabilidad de ser abordado para sufrir un robo o asalto (ver Fig. 3 y Tabla 3).

- Agentes guardianes o policías: están patrullando las calles en busca de agentes sospechosos para ser inspeccionados y posiblemente arrestados (ver Fig. 4 y Tabla 4).

3.1. Comportamiento de los agentes

La implementación de los algoritmos para que una situación de robo o asalto pueda ocurrir está basada en los diagramas de estado mostrados en las Fig. 2, Fig. 3 y Fig. 4. Estos algoritmos ofrecen la ventaja de dar la pauta de algunas de las causas y probabilidades que originan situaciones de robo y asalto en la vía pública, con la desventaja de que no se consideran factores sociales y cognitivos en las características de los agentes. Para que ocurra un robo o asalto deben cumplirse los siguientes hechos:

1. Un agente delincuente elige a un agente objetivo: al ocurrir este suceso el primero acelera su paso y empieza a seguir al segundo.
2. El agente delincuente se encuentra cerca del agente objetivo: dado este suceso, el primero está listo para atacar al segundo, pero esto sólo sucede cuando no hay otros agentes testigos potenciales dentro del radio de acción del primero.
3. El ataque se lleva a cabo: el agente delincuente roba o asalta al agente objetivo, el primero puede quedarse junto al segundo por algún tiempo para disfrutar de lo que ha hecho y luego se aleja para buscar un nuevo objetivo potencial. Sin embargo, si un agente delincuente no puede encontrar un buen momento para atacar a un agente objetivo, se da por vencido pasado un tiempo. Por último, la labor de los agentes guardianes es inspeccionar a cualquier agente ciudadano sospechoso, para arrestarlo en caso de que tenga un arma, o bien, se encuentre bajo el abuso de sustancias, de lo contrario, lo deja libre para seguir patrullando la calle.

```

//tell victim he is been attacked
isStopped = true;
if(randomValue < 0.2) {
  ((Victim)victim).statechart.receiveMessage("AggravatedAssault");
}
else if(randomValue < 0.9) {
  ((Victim)victim).statechart.receiveMessage("SimpleAssault");
}
else {
  abandonedAttack = true;
}

//tell victim he is been attacked
randomValue = Math.random();
if(randomTrue(decisionProbabilityOfPersonalTheft)) {
  if(randomValue < 0.6) {
    ((Victim)victim).statechart.receiveMessage("PersonalTheft");
  }
}

//just re-enter the state to move in a new direction
if(Math.random() < 0.05)
  isStopped = Math.random() < 0.5;
    
```

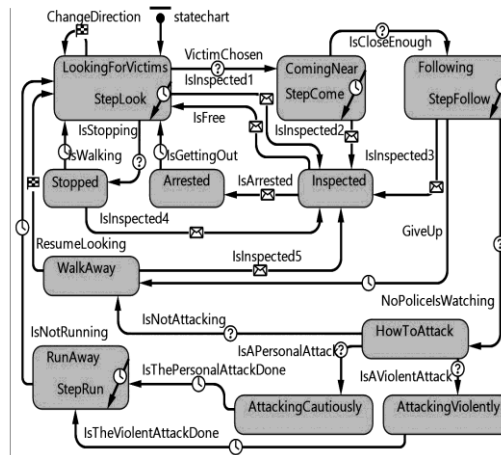


Fig. 2. Fragmentos del comportamiento de un agente delincuente y su diagrama de estados.

```

numberOfAttacksSuffered++;
get_Main().numberOfAttacks++;
if(gender.equals("female")) {
    get_Main().numberOfSilentAttacksAgainstFemales++;
}
isStopped = true;
isDistracted = false;
numberOfAttacksSuffered++;
get_Main().numberOfAttacks++;
if(gender.equals("female")) {
    get_Main().numberOfViolentAttacksAgainstFemales++;
}
if(msg.equals("AggravatedAssault")) {
    currentTheftSuffered = 1;
}
else if(msg.equals("Robbery")) {
    currentTheftSuffered = 2;
}
    
```

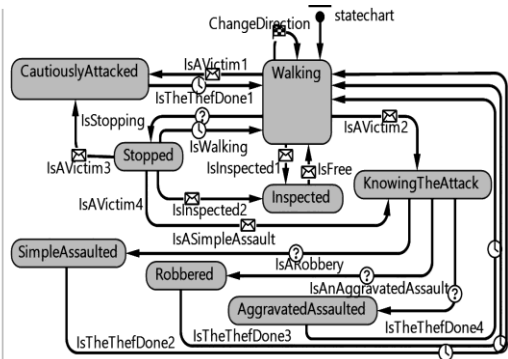


Fig. 3. Fragmentos del comportamiento de un agente objetivo y su diagrama de estados.

```

//tell suspicious that this is an inspection
if(suspicious instanceof Offender) {
    ((Offender)suspicious).statechart.receiveMessage("SubjectToInspection");
}
else {
    ((Victim)suspicious).statechart.receiveMessage("SubjectToInspection");
}

//tell suspicious that he is been arrested
((Offender)suspicious).statechart.receiveMessage("Arrested");

//tell suspicious that he is free
if(suspicious instanceof Offender) {
    ((Offender)suspicious).statechart.receiveMessage("Free");
}
else {
    ((Victim)suspicious).statechart.receiveMessage("Free");
}
    
```

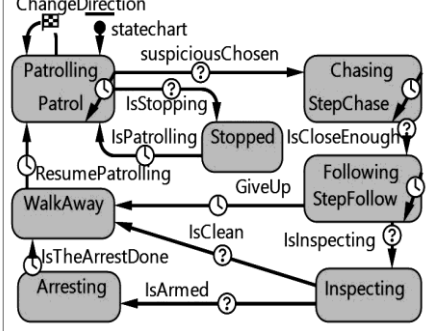


Fig. 4. Fragmentos del comportamiento de un agente guardián y su diagrama de estados.

3.2. Características de los agentes

Las principales características de los agentes presentes en el modelo propuesto se plasman en las siguientes tablas:

Tabla 2. Características del agente delincuente.

Nombre	Valores	Descripción
gender	{“male”, “female”}	Representa el género del agente, donde “male” es masculino y “female” indica femenino.
typePerson	[0,1)	Acorde a una división equitativa del rango de valores y del género, el agente puede tener una personificación de una enfermera, una persona ordinaria, un trabajador de la construcción, un empleado de oficina, un médico, un niño, una mujer rubia, una mujer morena, una niña.
hasAWeapon	{true, false}	El agente puede o no portar un arma para cometer un asalto.
isUnderSubstanceAbuse	{true, false}	Indica cuando el agente se halla bajo el efecto del abuso de alguna sustancia.
victim	Agent	Habla del agente objetivo que ha sido seleccionado para ser abordado.
isDistracted	{true, false}	El agente se halla distraído o no lo cual puede aumentar el riesgo de que sea una víctima a ser abordada.
isStopped	{true, false}	Estando detenido el agente puede ubicar a una posible víctima.

Tabla 3. Características del agente objetivo.

Nombre	Valores	Descripción
gender	{“male”, “female”}	Representa el género del agente, donde “male” es masculino y “female” indica femenino.
typePerson	[0,1)	Acorde a una división equitativa del rango de valores y del género, el agente puede tener una personificación de una enfermera, una persona ordinaria, un trabajador de la construcción, un empleado de oficina, un médico, un niño, una mujer rubia, una mujer morena, una niña.
CurrentTheftSuffered	[1,5]	Empleada para asignar el tipo de atraco que acaba de sufrir el agente, donde 1 es asaltado agravado, 2 es asalto intimidatorio, 3 es asalto simple, 4 es robo de cartera para un agente masculino y 5 es robo de bolso para un agente femenino.
hasMoreMoneyOrOther Belongings	{true, false}	Indica si el agente posee o no más dinero o pertenencias de las cuales sufra el riesgo de ser asaltado o robado.
isDistracted	{true, false}	El agente se halla distraído o no lo cual puede aumentar el riesgo de que sea una víctima a ser abordada.
isStopped	{true, false}	Estando detenido el agente se vuelve más vulnerable como posible víctima de robo o asalto.

Tabla 4. Características del agente guardián.

Nombre	Valores	Descripción
gender	{“male”, “female”}	Representa el género del agente, donde “male” es masculino y “female” indica femenino.
isDistracted	{true, false}	El agente se halla distraído o no lo cual puede aumentar el riesgo de que exista un robo o asalto.
isStopped	{true, false}	Estando detenido el agente puede inspeccionar el comportamiento de otros agentes.
suspicious	Agent	Corresponde al agente que es identificado como sospechoso

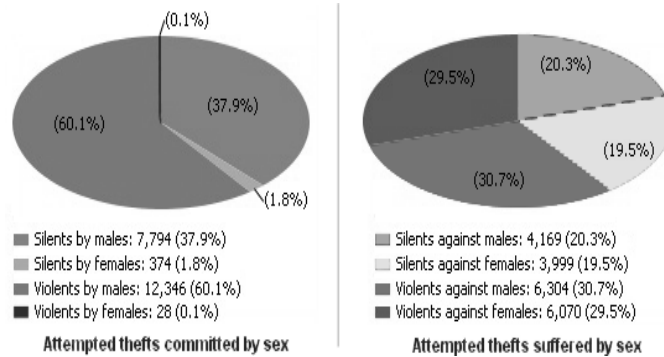


Fig. 5. Intentos de robo y asalto cometidos (izquierda) y sufridos (derecha) por sexo, sin agentes guardianes.

3.3. Casos de estudio

Para validar el modelo propuesto se corrió una simulación a un año para los siguientes casos de estudio:

1. Simulación sin agentes guardianes: en este primer escenario de simulación se contempló una población de agentes dada por: 2000 agentes objetivos, 995 del género femenino y 1005 del género masculino; 50 agentes delincuentes, 10 del

género femenino y 40 del género masculino, entre los cuales 1 mujer y 14 hombres fueron portadores de un arma, 1 mujer y 11 hombres estuvieron bajo el abuso de sustancias. Los resultados de esta simulación se muestran en la Fig. 5 y la Fig. 6.

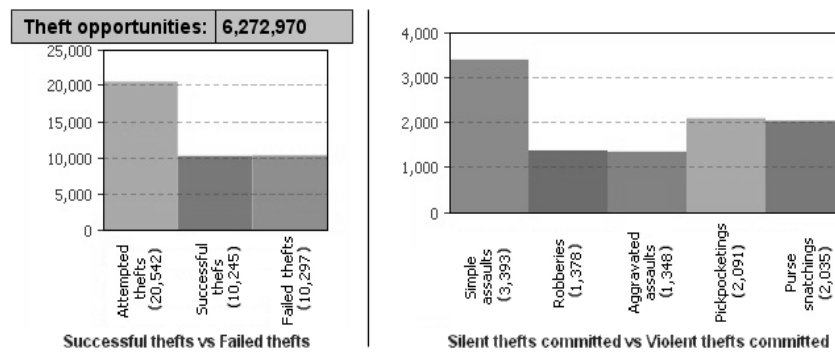


Fig. 6. Ataques exitosos vs fallidos (izquierda) y aquellos (derecha) que fueron asaltos (violentos) y robos (silenciosos), sin agentes guardianes.

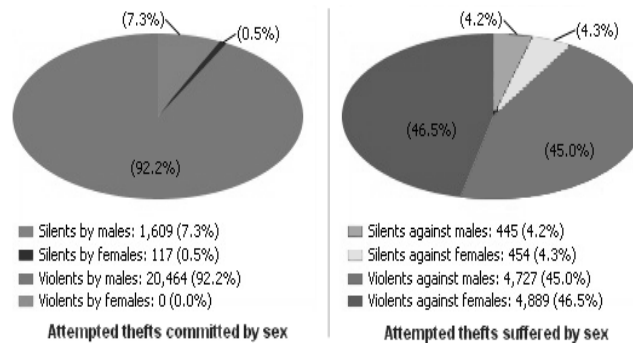


Fig. 7. Intentos de robo y asaltos cometidos (izquierda) y sufridos (derecha) por sexo, con agentes guardianes.

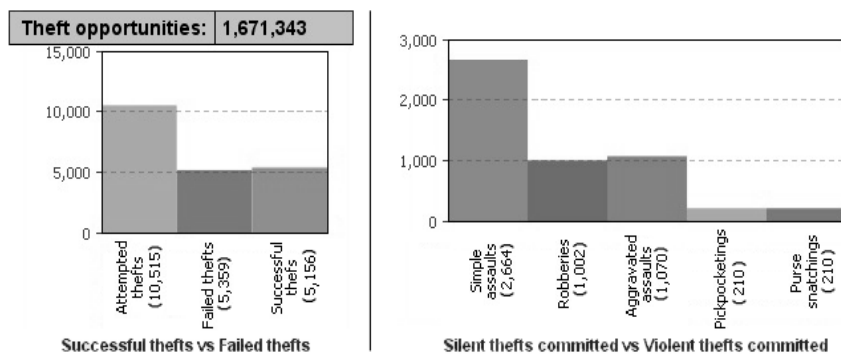


Fig. 8. Ataques exitosos vs fallidos (izquierda) y aquellos (derecha) que fueron asaltos (violentos) y robos (silenciosos), con agentes guardianes.

2. Simulación con agentes guardianes: para este segundo escenario de simulación se contempló una población de agentes dada por: 2000 agentes objetivos, 991 del género femenino y 1009 del género masculino; 50 agentes delincuentes, 19 del género femenino y 41 del género masculino, entre los cuales 7 hombres fueron portadores de un arma y 10 hombres estuvieron bajo el abuso de sustancias; 50 agentes guardianes, 8 del género femenino y 42 del género masculino, los cuales en total efectuaron 49,305 inspecciones a agentes sospechosos y sólo 164 de ellas fueron arrestos. Para esta simulación, los resultados que se obtuvieron se presentan en la Fig. 7 y la Fig. 8.

En la Fig. 5 y la Fig. 7 los intentos y ataques violentos corresponden a asaltos, mientras que los silenciosos indican robos.

4. Discusión de resultados

Cuando existe una población de agentes con igual número de delincuentes que de guardianes (ver Fig. 8), se registraron 1,671,343 oportunidades para intentar cometer ya sea robo o asalto. De esta cifra, se obtiene que sólo el 0.62% cayeron en intentos realizados, siendo el 49.03% exitosos y el 50.97% fallidos. De los crímenes que fueron exitosos, el 91.85% fue por asalto, teniendo al asalto simple como el de mayor incidencia.

En caso de tener una población de agentes en la cual no se cuenta con la presencia de algún guardián (ver Fig. 6), se observa que la cifra de oportunidades para intentar cometer robo o asalto casi se cuadruplica. Esta cifra fue de 6,272,970, de la cual se tiene que el 0.32% corresponde a intentos realizados, teniendo el 49.87% y el 50.13% como exitosos y fallidos, respectivamente. De la cifra de exitosos, el 59.72% fue por asalto, teniendo de nueva cuenta al asalto simple como tipo de crimen con mayor incidencia y el 40.28% por robo, teniendo ligeramente más robos de carteras que de bolsos.

5. Conclusiones y trabajo futuro

El modelado y la simulación basados en agentes han sido aplicados en diversas áreas del conocimiento, puesto que ofrecen la ventaja de poder reflejar de manera virtual a nivel individual o colectivo problemas y fenómenos sociales [17]. Sin embargo, en este artículo aquellos que abordan la delincuencia y la criminología fueron los que se tomaron como base para crear el modelo de simulación 3D presentado. Este modelo se distingue del resto en que se propone el uso de diagramas de estado para especificar y modelar comportamientos dinámicos y predefinidos con base en las características de los agentes creados. Estos comportamientos provocan que los agentes víctimas sufran un ataque según las probabilidades y oportunidades aprovechadas por los criminales.

Las principales ventajas que ofrece la implementación del modelo de simulación propuesto en AnyLogic son la recreación de un escenario realista 3D de un vecindario

en donde se puede observar a nivel individual el comportamiento de cada agente y a nivel estadístico la cantidad de oportunidades presentes durante el periodo de simulación para que ocurra un robo o un asalto y como la presencia de más agentes guardianes contribuye a que estas disminuyan y sea menor la cantidad de aquellos intentos que puedan resultar exitosos. Además, en la simulación se puede apreciar cómo es llevado a cabo el patrullaje para inspeccionar y arrestar, no obstante, aunque son muchas las inspecciones el número de arrestos es mínimo, lo cual dice que la manera de llevar a cabo esta actividad policiaca no ofrece muchas ventajas en la vida real y debe ser replanteada para un mayor rendimiento de los policías en las calles.

Es amplio el número de futuras aplicaciones basadas en la extensión y mejora del modelo propuesto. Estas aplicaciones pueden incluir la consideración de estrategias de predicción, prevención y reducción de los índices de robos y asaltos, por ejemplo al dar la pauta para que en el mundo real se lleven a cabo las prácticas puestas en marcha en los escenarios de simulación virtual.

Finalmente, se tiene contemplado el incluir el uso de las características cognitivas y sociales ligadas a situaciones de robo y asalto en los agentes, para tener un modelo más realista. Al igual, se pueden emplear en complemento con los diagramas de estado el uso de esquemas de inferencia tales como lógica difusa para especificar mediante reglas de inferencia las relaciones y situaciones entre tipos de robos y asaltos que puedan ocurrir. Dichas reglas evaluarían los rangos de pertenencia de todo factor involucrado, como lo son las características y estado de los agentes, así como del entorno simulado para determinar ante situaciones de incertidumbre cuando un robo o asalto es más probable que ocurra, o bien, una inspección se haga cuando en efecto un sospechoso tiene mayor certeza de ser un delincuente y deba ser arrestado. Además, para investigaciones posteriores se extenderá el tipo de crímenes o delitos modelados a otros derivados del fuero común [1], [2], [3].

Referencias

1. Secretariado Ejecutivo, <http://secretariadoejecutivo.gob.mx/index.php>
2. Incidencia Delictiva del Fuero Común 2015, http://secretariadoejecutivo.gob.mx/docs/pdfs/estadisticas%20del%20fuero%20comun/Cieisp2015_082015.pdf
3. Incidencia Delictiva Nacional, <http://secretariadoejecutivo.gob.mx/incidencia-delictiva/incidencia-delictiva-acumulado.php>
4. Bottoms, A. E., Claytor, A., Wiles, P.: Housing markets and residential community crime careers: A case study from Sheffield. In: Nicholas Fyfe, David Evans, and David Herbert, editors, *Crime, Policing and Place - Essays in Environmental Criminology*, chapter 7, Routledge (1992)
5. Brown, B. B., Bentley, D. L.: Residential burglars judge risk: The role of territoriality. *Journal of Environmental Psychology*, vol. 13, pp. 51–61, Elsevier (1993)
6. Baldwin, J. Bottoms, A. E.: *The Urban Criminal: A Study in Sheffield*. Tavistock Publications, London (1976)
7. Cohen, L. E., Felson, M.: Social Change and Crime Rate Trends: A Routine Activity Approach. *American Sociological Review*, vol. 44, pp. 588–608, Elsevier (1979)
8. Brantingham, P., Brantingham, P.: *Patterns in Crime*. The Journal of Criminal Law and Criminology, vol. 76, no. 2, pp. 540–543, New York: Macmillan (1984)

9. Cornish, D., Clarke, R. V.: *The Reasoning Criminal: Rational Choice Perspectives on Offending*. New York, NY: Springer-Verlag (1986)
10. Kongmuang, C.: *Modelling Crime: A Spatial Microsimulation Approach*. PhD thesis, University of Leeds, UK (2006)
11. Bonabeau, E.: Agent-based modeling: Methods and techniques for simulating human systems. In: *Proceedings of the National Academy of Sciences*, 99, pp. 7280–7287 (2002)
12. Canessa E., Quezada A.: *Modelamiento basado en agentes: un método de simulación computacional para la gestión de seguridad ciudadana*. Universidad Adolfo Ibañez, Chile (2013)
13. Devia N.: *Generación de datos de delincuencia vía simulación con modelos basados en agentes*. Facultad de Ciencias Físicas y Matemáticas, Departamento de ingeniería industrial, Universidad de Chile, Santiago Chile (2012)
14. Amrutha, S.: *Agent Based Simulation of Street Robbery*. Department of computer science, Royal college of engineering and technology Thrissur, Kerala, India (2014)
15. Malleson, N.: *Agent-Based Modelling and Crime in Leeds*. School of Geography, University of Leeds (2007)
16. AnyLogic: *Multimethod Simulation Software*, <http://www.anylogic.com/>
17. AnyLogic: *Application Areas*, <http://www.anylogic.com/application-areas>

Sistema de apoyo lingüístico en español para personas sordas

Obdulia Pichardo-Lagunas, Beatriz Torres-Alatraste, Bella-Citlali Martínez-Seis,
Víctor-Darío Cuervo-Pinto, Miguel-Alejandro Martínez-Rosales

Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, D.F.,
México

therasmus_bet@hotmail.com,
{opichardola, vdcuervo, bmartinezs, mamartinezr}@ipn.mx

Resumen. Se propone un sistema computacional que sirve como herramienta de apoyo a personas sordas en el aprendizaje de la lengua española escrita. La aplicación desarrollada utiliza herramientas de Procesamiento de Lenguaje Natural para el análisis y validación de los resultados obtenidos en los problemas propuestos. Los dos ejercicios planteados en el sistema tienen como fin desarrollar habilidades de escritura de la lengua española en las mencionadas personas. En el primer caso se implementa la validación sintáctica en la construcción de oraciones con unidades léxicas previamente definidas y en el segundo el análisis léxico y sintáctico en la escritura de oraciones relacionadas a tópicos específicos planteados por especialistas en la enseñanza a personas sordas.

Palabras clave: Sordo, procesamiento de lenguaje natural, lingüística computacional, sistema computacional, analizador sintáctico, detección de errores.

System for Linguistic Support in Spanish for Deaf Persons

Abstract. We propose a computational system to be used by deaf people as a tool in their written Spanish language learning. Natural Language Processing techniques were used to build the computer program in order to analyze and validate obtained results in the offered exercises. Two problems are offered in the computer program. Both of them are intended to improve Spanish writing abilities in deaf people. A syntactic validation is carried out in the first proposed exercise. This is done when the user builds sentences in written Spanish. Those problems use a previously defined lexicon. In the second kind of problems a lexical and syntactic analysis is performed. The lexicon and the syntax are related to specific topics in Spanish teaching for deaf people.

Keywords: Deaf, natural language processing, computational linguistics, computer system, parser, error detection.

1. Introducción

El presente trabajo se centra en el desarrollo de una herramienta para el uso de las personas hipoacústicas. En el año 2010, las personas que tuvieron algún tipo de discapacidad fueron 5 739 270 en México. Las personas con limitación para escuchar fueron un número de 694 464 [1]. Por los motivos anteriores se considera pertinente el desarrollo de herramientas para su apoyo.

a. Problemática

Las lenguas humanas constituyen entes de gran magnitud. Por ello requieren, en ocasiones, acercamientos de orden estadístico para su estudio [2]. Se decidió enfrentar el problema de la magnitud de la lengua mediante el uso de una sintaxis y un léxico reducidos.

Un lenguaje formal es un conjunto de cadenas de símbolos sobre un alfabeto, al que se denomina vocabulario. Los lenguajes que usan las personas para su comunicación interpersonal son llamados lenguajes naturales. Una de las diferencias entre los lenguajes naturales y los formales es la complejidad de sus estructuras de conocimiento. El tratamiento de esta complejidad requiere del uso de formalismos específicos para la representación de conocimiento lingüístico, así como algoritmos y técnicas computacionales adecuadas [3].

El problema a resolver en este artículo resulta en desarrollar un sistema que proporcione funciones de análisis léxico y sintáctico de oraciones en español para su uso por personas sordas. Los usuarios deben conocer el abecedario, algunas palabras y algunas estructuras gramaticales del español.

b. Marco teórico

El Procesamiento de Lenguaje Natural (PLN) es el reconocimiento y utilización de la información expresada en un lenguaje humano a través de sistemas informáticos [4]. El PLN comprende diferentes niveles de análisis, a saber el morfológico, el sintáctico, el semántico y el pragmático [5]. Los formalismos lógicos permiten dichos análisis. Dentro de los diferentes problemas que se presentan en el procesamiento del lenguaje, el de la ambigüedad es el principal. Éste aparece cuando alguna unidad lingüística, a saber, sonido, palabra, oración, se puede interpretar en más de una manera [5]. Tal situación se puede notar, a nivel morfológico, por ejemplo, en la palabra “traje” en la oración “traje bebidas” a diferencia de la palabra con la misma forma en la oración “viste de traje”.

El PLN incluye a los sistemas traductores automáticos. Mediante ellos, un usuario puede leer, en su propio lenguaje, un texto escrito en otro lenguaje. Asimismo puede escribir en su lenguaje para otros que usan un lenguaje diferente. También es capaz de conversar, de forma escrita u oral, con otros que no comparten su misma lengua [2].

Los lenguajes formales pueden usarse para modelar lenguajes naturales. Para generar lenguajes formales se puede utilizar un procedimiento llamado gramática libre de contexto (GLC), que es equivalente a una forma Backus-Naur (*BNF* por sus siglas en inglés). Una GLC es un conjunto de reglas de “producciones”, cada

una de las cuales expresa las maneras en que los símbolos del lenguaje pueden ser agrupados y ordenados unos junto a otros; además de un vocabulario o diccionario de palabras y símbolos. Formalmente, una GLC es una 4-tupla que puede representarse genéricamente como $G = (N, \Sigma, R, S)$. Aquí, N es un conjunto de símbolos no terminales, Σ es un conjunto de símbolos terminales, R es un conjunto de reglas de producción, S es el símbolo inicial de la gramática. El conjunto de reglas de producción está formado por expresiones como $A \rightarrow \beta$. Ésta se puede leer “ A deriva en β ”. Aquí A es la “cabeza de la producción” o “antecedente”. Además, en la expresión $A \rightarrow \beta$, el símbolo A es no terminal. β es una cadena de símbolos del conjunto infinito de cadenas $(\Sigma \cup N)^*$ [6].

Un lenguaje se define a partir de la “derivación”. Una cadena deriva en otra si puede ser reescrita como la segunda mediante la aplicación de una serie de reglas. Formalmente podemos establecer que si $A \rightarrow \beta$ es una producción de P y α y γ son cualquier cadena en el conjunto $(\Sigma \cup N)^*$, entonces decimos que $\alpha A \gamma$ “deriva directamente” en $\alpha \beta \gamma$, o bien que $\alpha A \gamma \Rightarrow \alpha \beta \gamma$. Así, la derivación es una generalización de la derivación directa:

Sean $\alpha_1, \alpha_2, \dots, \alpha_m$ cadenas en el conjunto $(\Sigma \cup N)^*$, con $m \geq 1$ tal que $\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \dots, \alpha_{m-1} \Rightarrow \alpha_m$, entonces decimos que α_1 deriva en α_m , o bien, $\alpha_1 \Rightarrow^* \alpha_m$.

Formalmente se puede definir un lenguaje \mathcal{L}_G generado por una gramática G como el conjunto de cadenas compuestas por símbolos terminales que pueden ser derivados de un cierto símbolo inicial S . Esto es $\mathcal{L}_G = \{w | w \text{ está en } \Sigma^* \text{ y } S \xRightarrow{*} w\}$. Note que en un vocabulario, el símbolo no terminal asociado con cada palabra define una categoría léxica, la categoría de la palabra (sustantivo, artículo, adjetivo, verbo, etc.) [6]. Un componente léxico o *token* es un elemento de una categoría léxica (perro, los, alto, reír, etc.).

Las gramáticas regulares son equivalentes a las expresiones regulares. Las gramáticas regulares pueden ser “*left-linear*” o “*right-linear*”. Una regla en una gramática *right-linear* tiene un solo no terminal en la izquierda y a lo más un no terminal en el lado derecho. Si existe un no terminal en el lado derecho, debe ser el último símbolo de la cadena. El lado derecho de una gramática *left-linear* tiene una configuración contraria (el lado derecho debe empezar con máximo un único no terminal). Todos los lenguajes regulares tienen gramáticas *left-linear* y *right-linear* [6].

Es posible realizar diferentes tipos de análisis en PLN, como el morfológico, el sintáctico, el semántico, de discurso, etc. El problema de reconocer que una palabra, por ejemplo, en plural, digamos *zapatos* se puede descomponer en morfemas (“zapato” y “-s”) y construir una representación estructurada de tal descomposición, se conoce como análisis morfológico. Un análisis sintáctico o *parsing* es la combinación del reconocimiento de una cadena (oración) de entrada con la asignación a ella de una estructura sintáctica. Se suele representar tal estructura (o derivación) mediante un árbol. Tales árboles comúnmente se muestran invertidos, es decir la raíz en la parte superior. Se necesita especificar los algoritmos mediante los cuales las gramáticas libres de contexto producen los árboles. Un analizador sintáctico o *parser* puede ser visto como como una búsqueda por el espacio de posibles árboles hasta encontrar el árbol correcto que reproduzca una oración dada. La mayor parte de los

analizadores sintácticos utilizan dos estrategias, la búsqueda descendente o búsqueda dirigida por objetivo y la búsqueda ascendente o búsqueda dirigida por datos. La búsqueda descendente recorre un árbol sintáctico de la raíz a las hojas. Por su parte la ascendente lo hace de las hojas a la raíz [6].

Existen herramientas para el desarrollo de analizadores sintácticos como BISON, YACC (Yet Another Compiler-Compiler) y JavaCC (Java Compiler Compiler). Los dos primeros usan salidas en código C y C++ respectivamente, mientras que JavaCC es de código abierto para el lenguaje de programación Java. JavaCC genera analizadores descendentes, permite la construcción de árboles de abajo hacia arriba, hace derivaciones por la izquierda, usa gramática LL(1) y LL(k), una versión de EBNF (Extended Backus-Normal Form), proporciona integración de los analizadores léxicos y sintáctico, permite el uso de *tokens* especiales (a ser procesados por el desarrollador). Se halla entre las que tienen mejor manejo de errores, además de ser una herramienta de nueva generación.

2. Metodología

El sistema es una herramienta de apoyo para el proceso de aprendizaje lingüístico para las personas sordas. Con el fin de identificar los elementos sobresalientes del lenguaje español utilizado en este tipo de enseñanza se realizó una investigación y así se obtuvieron las palabras más usadas. Con ello se definió un determinado léxico. De la misma manera se consultaron, y posteriormente se definieron las estructuras gramaticales más usadas por dicho sector de la población.

El sistema consta de dos módulos, un cliente y un servidor cuya arquitectura se muestra en la Fig. 1.

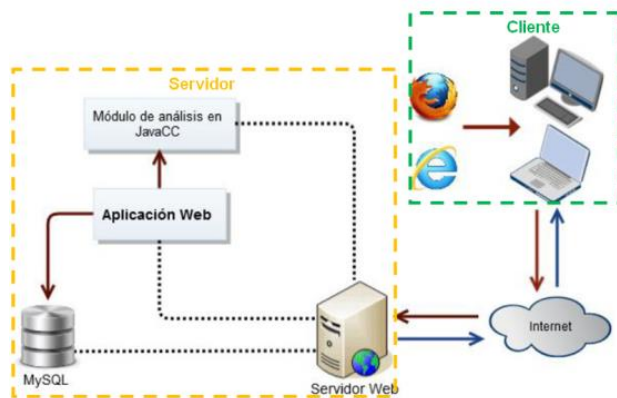


Fig. 1. Arquitectura de alto nivel para el sistema de apoyo lingüístico.

En el servidor se implementa un analizador léxico y uno sintáctico. El primero consiste en validar que las palabras utilizadas se encuentren previamente definidas en una base de datos. Por su parte, el segundo es el que valida que se estén usando, también, sólo las normas de gramática predefinidas. La relación entre los analizadores léxico y sintáctico se esquematiza en la Fig. 2.

El cliente contiene el despliegue de la interfaz gráfica a través del navegador *Web*, en la cual puede tener acceso a dos ejercicios que ayudan a su desarrollo lingüístico. Ambos tipos ejercicios realizan un análisis sintáctico, y uno de ellos también un análisis léxico. El sistema permite al usuario la corrección de la estructura que ha escrito en español, si ésta contiene algún error, hasta, eventualmente obtener una construcción adecuada.

Para comunicar ambos módulos se usa el *framework* Java Server Faces ya que permite separar la vista, o presentación, de la lógica de negocio mediante su patrón de diseño MVC2 (Modelo Vista Controlador). Es definido como un *framework* extensible.

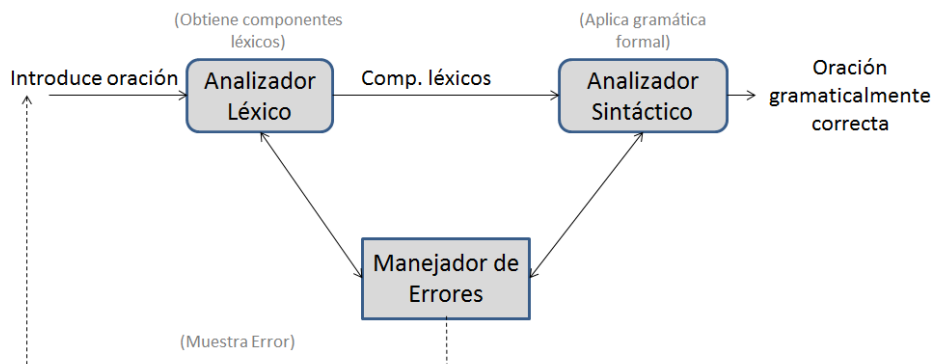


Fig. 2. Método de validación gramatical del sistema.

A continuación describiremos con mayor detalle a los módulos de servidor y de cliente.

Servidor. El servidor se encarga del procesamiento del analizador léxico y sintáctico haciendo uso de JavaCC por sus características lexicográficas y sintácticas requeridas para el análisis de oraciones. Así mismo, se encarga de procesar las llamadas y conexiones al gestor de base de datos MySQL, que es el encargado de almacenar las oraciones y las imágenes que conforman los dos tipos de ejercicios.

Implementación de la gramática formal. Se diseñó una gramática formal a partir del conjunto de estructuras oracionales y vocabulario restringido que, como ya se mencionó, fueron definidas por la investigación. Con el fin de implementar dicha gramática en forma computacional, se utilizaron diferentes herramientas.

JavaCC permite realizar un análisis descendente (genera el árbol sintáctico correspondiente desde la raíz a las hojas) y hacer derivación por izquierda (gramática LL). Para el desarrollo de la gramática LL se hizo uso de la notación EBNF por ejemplo:

```
<forma nominal> := <articulo><sujeito> | <articulo><sujeito><preposicion>
```

Para modelar la gramática se generaron de manera individual los árboles sintácticos de cada una de 15 estructuras y se constituyeron en uno solo. Recuérdese que un sintagma es una unidad lingüística compuesta por una o más palabras a la que también se le ha llamado “frase”. En este sistema se utilizan 5 sintagmas principales **sintagma nominal** cuyo núcleo es el sustantivo o pronombre, **sintagma verbal** con

un verbo como núcleo, **sintagma adverbial** (adverbio como núcleo), **sintagma adjetival** (adjetivo como núcleo) y **sintagma preposicional** (preposición como núcleo). La conjunción sólo realiza la función de enlace, siempre entre sintagmas del mismo tipo. Cada uno de ellos se deriva en componentes más pequeños, e incluso en otros sintagmas, que también cumplen una función específica dentro de una oración obteniendo el árbol que se muestra en la Fig. 3.

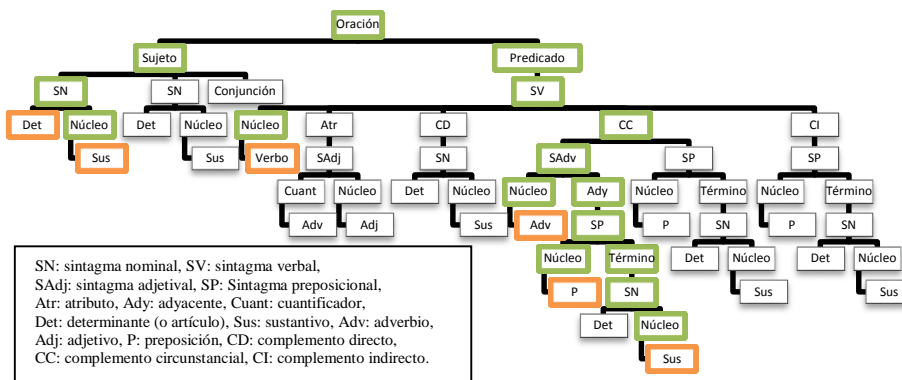


Fig. 3. Representación de la gramática mediante una estructura de árbol.

Analizador léxico. En la especificación léxica, mediante el uso de expresiones regulares, se valida que todas las palabras que conforman la oración correspondan con el vocabulario definido. Se manejan básicamente tres elementos:

- Patrones: las reglas que describen un conjunto de caracteres o palabras.
- Componentes léxicos o *tokens*: símbolos terminales de la gramática.
- Atributos: información adicional para cada componente léxico (relevante para análisis semántico).

En este proceso se leen los caracteres de entrada para verificar un patrón, se obtiene el *token* (carácter a carácter) y se espera la petición del analizador sintáctico para entregar el *token* que ya constituye una palabra.

Configuración léxica. Debido a la delimitación del vocabulario se definió directamente la morfología de cada una de las palabras, considerando sus accidentes gramaticales: género y número. Por ejemplo para artículos tenemos la siguiente definición del léxico mediante sintaxis JavaCC:

```
<ARTICULO:    ("E"|"e") ("l")      |    ("L"|"l") ("a")      |    |
              ("L"|"l") ("os")   |    ("L"|"l") ("as")   |    ("U"|"u") ("n")   |
              ("U"|"u") ("na")   |    ("U"|"u") ("nos")   |    ("U"|"u") ("nas") >
```

Se analiza el flujo de caracteres clasificándolos en *tokens* (palabras válidas) y asignándoles un componente léxico, verificando el máximo número de caracteres posibles que correspondan a una palabra definida en la sintaxis léxica. Se asignan

valores enteros para cada componente léxico al igual que el atributo `image` mediante el cual se vincula la cadena analizada a un componente definido. En caso contrario se genera y despliega un error.

Analizador sintáctico. Con la especificación sintáctica se corrobora que la cadena de entrada cumpla con una determinada estructura. Es así que es posible identificar si la oración de entrada corresponde a cualquiera de las estructuras definidas. Esto involucra la petición al analizador léxico, examinar el símbolo de la regla, aplicar la derivación y verificar la correspondencia del *token*.

Configuración sintáctica. Se realizó la definición de las reglas gramaticales mediante el formato *EBNF* propio de Java CC, compuesta por elementos terminales (como el espacio, *SPACE*, en el siguiente ejemplo) y elementos no terminales (como el predicado en el siguiente ejemplo) que son derivados hasta encontrar el símbolo final. Veamos el ejemplo:

```
void oración():{  
{  
sinNominal() <SPACE> ( predicado() | suj_compuesto()  
predicado() | sinPreposicional() <SPACE> predicado() )  
}}
```

Las estructuras validadas por el analizador son aquellas definidas por la gramática previamente expuesta. Cuando no se encuentra una correspondencia válida se arroja una excepción que indica el error.



Fig. 4. Ejemplo de ejercicio “Formar Oración”.

Cliente. El sistema proporciona al usuario con discapacidad auditiva la selección de dos ejercicios que apoyan su aprendizaje del lenguaje español, los cuales están disponibles en cualquier tipo de navegador Web con conexión al servidor.

Existen dos tipos de actores: el usuario que interactúa con los ejercicios del sistema y el administrador que se encarga de la edición de los ejercicios. Para diferenciar el tipo de actor se ha creado un Inicio de sesión que permite dar los permisos respectivos a cada uno de los usuarios.

Ejercicio “Formar Oración”. El primero de los tipos de ejercicios consiste en arrastrar varias cajas de texto hacia un contenedor. Al usuario se le presenta una oración en desorden que cumple con alguna de las 15 estructuras gramaticales definidas (Ver Fig. 4). Dicha oración se encuentra almacenada en la base de datos del servidor, al que se accede de manera remota. Del mismo modo, a través del uso de *servlets*, se hace la comunicación para la verificación de la oración en el analizador léxico-sintáctico del servidor.

Ejercicio “Escribir Oración”. El segundo tipo de ejercicio consiste en redactar una oración relacionada con la imagen mostrada. Al usuario se le despliega de manera aleatoria una de las imágenes almacenadas en el servidor, posteriormente debe ingresar una oración relacionada con la imagen, por lo que se permite un uso restringido de caracteres. Una vez ingresada la oración se envía al analizador en el servidor para su análisis, en caso de ser correcta se muestra el acierto como en la Fig. 5 o el respectivo error para una oportunidad más.



Fig. 5. Ejemplo de ejercicio “Escribir Oración”.

Módulo de edición. El usuario administrador puede crear, eliminar o modificar algún ejercicio. En el tipo de ejercicio “Formar Oración”, el usuario puede ingresar nuevas oraciones o seleccionar alguna de las existentes para su edición o borrado. Del mismo modo el administrador podrá realizar dichas acciones para las imágenes requeridas en el segundo tipo de ejercicio. En ambos casos las modificaciones se realizan desde el navegador del cliente y quedan respaldadas en el servidor.

3. Resultados

En esta sección se muestran los resultados obtenidos en la evaluación durante la ejecución del sistema. Los ejercicios a validar son: Formar oración y Escribir oración.

En el ejercicio "Formar oración" se muestran las palabras desordenadas de manera aleatoria. Supongamos que las palabras presentadas son:

debajo	de	Las	cama	están	pelotas	la
--------	----	-----	------	-------	---------	----

una oración que el sistema valida como correcta es:

Las	pelotas	están	debajo	de	la	cama
-----	---------	-------	--------	----	----	------

ya que corresponde a la estructura gramatical:

Art - Sus - Verb - Adv - Prep - Art – Sus.

Esta estructura tiene el subárbol sintáctico que se muestra de forma resaltada en la Fig. 3.

El sistema no realiza la validación semántica por lo que no estudia el significado de la expresión lingüística. Tomando como válida la siguiente estructura que corresponde a la misma estructura gramatical:

Las	cama	están	debajo	de	la	pelotas
-----	------	-------	--------	----	----	---------

Para el mismo conjunto de palabras el usuario podría introducir la oración:

Las	pelotas	están	debajo	cama	la	de
-----	---------	-------	--------	------	----	----

En este caso la palabra "cama" se encuentra ubicada erróneamente dentro de la estructura de la oración, por lo que se marca en rojo con un texto descriptivo en la aplicación. El texto en el sistema señala para este caso, que "CAMA" es un sustantivo y que las opciones esperadas son una preposición o contracción según las estructuras gramaticales definidas.

Es necesario que el usuario identifique qué elemento de la oración corresponde a cada componente, es decir, de la oración que se muestra, cuál palabra representa un sustantivo, cuál un verbo, cuál un adjetivo, etc.

En la Fig. 4 se observa que las palabras presentadas son:

profesores	felices	bailan	Los
------------	---------	--------	-----

El usuario arrastra cada una de las palabras y puede formar la oración:

Los	profesores	bailan	felices
-----	------------	--------	---------

la cual corresponde a una estructura gramatical: Art-Sus-Verb-Adv, la cual es válida. Para este mismo conjunto de palabras el usuario puede ingresar la oración:

bailan	felices	Los	profesores
--------	---------	-----	------------

la cual presenta una estructura : Verb-Adv-Art-Sus que según el árbol sintáctico expuesto en la Fig. 4 es una estructura válida, a pesar de que la palabra "LOS" está en mayúscula ocupando el tercer puesto.

Los usuarios de la aplicación son sordos que ya han sido alfabetizados, por lo que deberían conocer los componentes léxicos y su función en la oración, lo que practicarán con la herramienta será manejarlos en conjunto.

Para el segundo tipo de ejercicio, previo al análisis sintáctico antes descrito, se lleva a cabo el análisis léxico. El usuario ingresa una oración relacionada con la imagen. Suponga que para la imagen de la Fig. 5 el usuario ingresa la oración: "**El sñor está enojado**" el sistema marcaría que el formato no es válido, ya que únicamente se permite el uso de caracteres. Si se introduce algún dígito o signo ortográfico no permitido (a excepción de acentos) se notifica al usuario, y por lo tanto no se procesa la solicitud de enviar la oración al analizador para su análisis.

Para la oración "**El señor está enojdo**" se puede observar que las palabras "**señor**" y "**enojdo**" están mal escritas, en este caso el sistema marcaría en naranja "señor" diciendo que dicha palabra no existe o está escrita incorrectamente.

En caso de que el análisis léxico sea correcto se manda a llamar al analizador sintáctico. Por ejemplo para la oración "**El señor está muy enojado**", el análisis léxico para cada una de las palabras es satisfactorio, posteriormente se procede al análisis sintáctico detectando la estructura gramatical:

Art-Sus-Verb-Adv-Adj.

Los análisis en la aplicación se realizan de forma secuencial por lo que si el sistema detecta un error lo reporta y detiene el proceso. Si existen dos errores, las sugerencias de corrección solo son con respecto al primer error presentado.

4. Conclusiones

La aplicación detecta los errores a nivel léxico y sintáctico, pero sólo considerando una gramática específica. La validación de oraciones gramaticalmente correctas se realiza satisfactoriamente, sin embargo en algunas ocasiones, tales oraciones, carecen de sentido, por lo que la inclusión del análisis semántico mejoraría el trabajo presentado.

A pesar de que las estructuras gramaticales propuestas eran sencillas, al probarlo con personas con discapacidad auditiva se corroboró la dificultad que para ellos representa expresarse en español escrito; por lo que se podría mejorar contando con diversos niveles de dificultad, así como la inclusión de videos que expliquen con lengua de señas mexicana el ejercicio a realizar.

Referencias

1. Censo de Población y vivienda (Cuestionario ampliado). Instituto Nacional de Estadística, Geografía e Informática (INEGI), <http://www3.inegi.org.mx/sistemas/TabuladosBasicos/Default.aspx?c=27303&s=est> (2010)
2. Gelbukh, A.: Procesamiento de lenguaje natural y sus aplicaciones. *Komputer Sapiens*, Vol. 1, pp. 6–11 (2010)
3. Quesada Moreno, J.F., de Amores Carredano, J.G.: Diseño e implementación de sistemas de traducción automática. Universidad de Sevilla, Sevilla (2000)

4. Alfonso Galipienso, M.I., Cazorla Quevedo, M.A., Colomina Pardo, O., Escolano Ruiz, F., Lozano Ortega, M.Á.: *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Paraninfo, Madrid Tesis (2003)
5. Gelbukh, A., Sidorov, G.: *Procesamiento automático del español con enfoque en recursos léxicos grandes*. Dirección de Publicaciones del IPN, México (2010)
6. Jurafsky, D., Martin, J.H.: *Speech and Language Processing, an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, New Jersey (2008)

Sistema de monitoreo remoto de sensores de temperatura y corriente usando una red híbrida bus CAN–Zigbee

Héctor Anzures Martínez, Ángel Bulmaro Sánchez Rubio, Josefina Castañeda Camacho, Gerardo Mino Aguilar, José Fermi Guerrero Castellanos, Liliana Cortez

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Electrónica, Puebla,
México

Resumen. Este trabajo trata sobre un sistema de monitoreo remoto a través de una pantalla inteligente para sensores de temperatura y corriente utilizando una red híbrida CAN–ZIGBEE. El CAN bus es usado como medio de transmisión de datos a corta distancia mientras que Zigbee es empleado para que cada nodo de la red pueda interactuar de manera inalámbrica con el nodo principal. De esta manera la red híbrida combina las ventajas de cada protocolo de comunicación para intercambiar datos. El sistema cuenta con cuatro nodos, dos son CAN y reciben la información de los sensores y el resto son Zigbee. Estos nodos están a cargo de transmitir la información de un nodo CAN de manera inalámbrica y desplegarla en una pantalla inteligente.

Palabras clave: CAN bus, Zigbee, dsPIC, monitoreo, pantalla inteligente.

System for Remote Monitoring of Temperature and Current Sensors Using Hybrid Red Bus CAN-Zigbee

Abstract. This paper describes a remote monitoring system that uses an intelligent screen with temperature and current sensors and a CAN bus-Zigbee hybrid network. CAN bus is used as short distance data transmission medium, while Zigbee is used in a way that every node in the network can interact wirelessly with the main node. In this way, the network combines the advantages of every single communication protocol for data interchange. The system is composed by 4 nodes, 2 nodes are CAN and they receive the sensors' information and the rest of the nodes are Zigbee. These nodes are responsables of transmitting information of a CAN node wirelessly and display it on an intelligent screen.

Keywords: CAN bus, Zigbee, dsPIC, monitoring, intelligent screen.

1. Introducción

En la actualidad el uso de sistemas de monitoreo para distintas aplicaciones es importante debido a que proporcionan al usuario una interfaz para poder visualizar el funcionamiento de dicho sistema y detectar posibles errores. Un punto esencial para estos sistemas es la obtención de los resultados de manera sencilla. El bus CAN es un protocolo de comunicación serial con topología de bus diseñado para el intercambio de información entre múltiples unidades de control electrónico dentro de un sistema distribuido, las principales ventajas de este protocolo son: robustez ante interferencias electromagnéticas, intercambio de información en tiempo real, transmisión de hasta 8 bytes por trama, estandarizado, topología bus que reduce considerablemente el cableado, etc. Las características que presenta el CAN Bus lo han ubicado como protocolo de facto en la industria automotriz y la industria de la automatización [1].

La tecnología Zigbee fué creada por la Zigbee Alliance en el 2004 [2]. Esta tecnología representa una opción atractiva dentro de las redes WPAN (Red de Área Personal) [3,4]. Zigbee se basa en el protocolo IEEE 802.15.4 y es un tópico importante en la investigación de tecnologías de comunicación inalámbrica de corto alcance. Sus principales ventajas son el bajo consumo de potencia, auto organización de baja complejidad y su bajo costo. Es ampliamente utilizada en la automatización de edificios, monitoreo y control de hospitales y hogares, a corta distancia. En este trabajo se describe el diseño e implementación de un sistema de monitoreo de sensores de corriente y temperatura a través de una red híbrida bus CAN–Zigbee.

2. Diseño del sistema

2.1. Diagrama a bloques del sistema

La Figura 1 muestra el diagrama a bloques del sistema de monitoreo integrado por 4 nodos. Cada nodo está formado por un dsPIC30F3011 y un transceptor CAN MCP2551. El nodo 1 lee la información del sensor 1 (sensor de corriente). Mientras que el nodo 2 lo hace de un sensor de temperatura. Ambos nodos envían información hacia el nodo central. De aquí se envía a un transmisor Zigbee que lo transmite inalámbricamente hacia su correspondiente receptor Zigbee. Éste dispositivo está conectado vía serial con una pantalla LCD donde se despliega la información de los sensores de la red.

2.2. Red CAN

Los nodos de la red CAN [5] están integrados por el dsPIC30F4013 el cual contiene un módulo CAN en su arquitectura para poder establecer comunicación

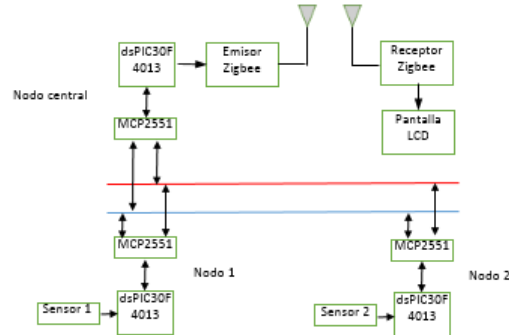


Fig. 1. Diagrama a bloques del sistema.

con otros dispositivos y el transceptor MCP2551 encargado de convertir los niveles de tensión entregados por el dsPIC a los niveles de tensión que necesita una red CAN bus. La Figura 2 muestra la topología utilizada.

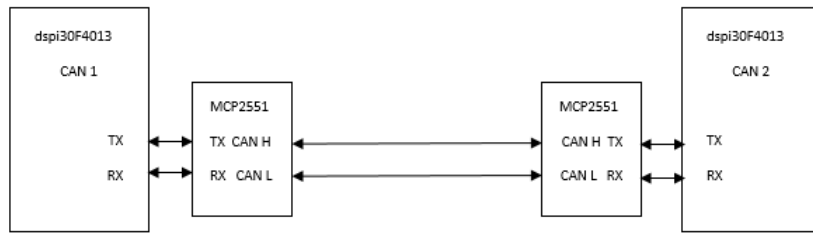


Fig. 2. Topología de red CAN.

2.3. Matriz CAN

Para la red CAN se diseñó la matriz de comunicación, en esta matriz se especifica los mensajes que se transmitirán a través de la red, ID, DLC y señales que transportarán las tramas. La Tabla 1 muestra la información de esta matriz.

2.4. Red Zigbee

Los módulos Zigbee utilizados, han sido diseñados para cumplir con la norma IEEE 802.15.4. Lo que significa bajo costo, bajo consumo de energía y seguridad. Este dispositivo posee 20 pines y opera dentro de la banda de frecuencia ISM de 2.4 GHz. Tiene la capacidad de transmisión de datos 100 metros con una tasa de 250.000 bps. Los módulos Zigbee operan en cinco modos. Cuando no

Tabla 1. Matriz de comunicación CAN.

Id	Tx	Rx	Dlc	Señal	Bit de inicio	Bits de la señal	Valor mínimo	Valor máximo	Interpretación
069	CAN1	Nodo central	4	Temperatura y corriente	0	15	0	255	Valor digital
073	CAN2	Nodo central	4	Temperatura y corriente	0	15	0	255	Valor digital

reciben o transmiten datos, el módulo de RF está en modo inactivo. El módulo de RF cambia a los otros modos de operación bajo diversas condiciones. En el modo de transmisión de datos en serie los datos son recibidos en el pin DI (Data In) y se almacenan en el buffer DI hasta que puedan ser procesados. Cuando el buffer DI llega 17 bytes, de forma predeterminada, el pin CTS pasa a estado lógico alto para indicar al dispositivo host que detenga el envío de datos. En modo recepción, los datos se reciben a través de la antena. Cuando un dato RF es recibido entra por DO (Data Out). La red Zigbee que se utiliza consta de 2 dispositivos que establecen una comunicación punto a punto. Se hace uso de módulos XBEE serie 2 [6] para implementar esta red. La Figura 3 muestra la red Zigbee que se utiliza.



Fig. 3. Diagrama de bloques de la red Zigbee.

2.5. Interfaz de usuario

La Interfaz de Usuario consiste en la forma en que el usuario interactúa con el sistema de monitoreo de sensores. La forma más simple es haciendo uso de una pantalla táctil en la que se muestren la información que recolectan los sensores. El diseño de la Interfaz de Usuario se ajusta a una pantalla específica que en este caso corresponde al modelo uLCD-70DT del fabricante 4D Systems.

Se trata de una pantalla de 7 pulgadas con las siguientes características:

- Procesador gráfico DIABLO16.
- Puerto de entradas y salidas de propósito general
- Niveles de voltaje TTL.

Para diseñar la interfaz se requiere el uso del software Workshop creado por el mismo fabricante. Este software posee 4 modos de programación de sus pantallas.

En este trabajo se utiliza el modo Visi-Genie ya que la idea de la interfaz es que sea lo más simple y atractiva para que el usuario pueda interactuar rápidamente con ella. Este modo de programación nos brinda un estilo gráfico a base de botones, switches, indicadores, etc.

El flujo del diseño de la Interfaz de Usuario obedece a la Fig. 4 . Al inicio se selecciona el modelo de pantalla para que el software ajuste la ventana de diseño al tamaño real de la pantalla, luego se escoge el modo de programación de ésta, en nuestro caso es Visi-Genie. Continuamos con la configuración de los pines del puerto GPIO como entradas o como salidas según se requiera. Luego se agregan los botones e indicadores que servirán para mostrar el estado de los sensores de temperatura y corriente. Posteriormente se asocia cada botón e indicador a un pin del puerto GPIO para que responda a cada evento que el usuario ordene. Finalmente, el puerto serial lee señales digitales analógicas provenientes de los sensores.

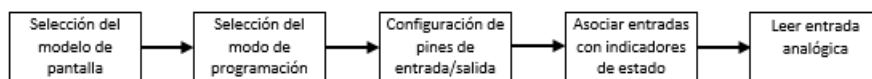


Fig. 4. Flujo de diseño.

2.6. Sensor de corriente

El sensor de corriente que se utiliza es el modelo ACS712ELC-30A del fabricante Allegro. Este sensor realiza el sensado de corriente mediante efecto hall [7]. En la Figura 5 se muestra el esquema de conexión de este sensor.

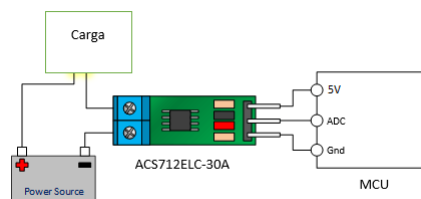


Fig. 5. Esquema de conexión del sensor ACS712ELC-30A.

Este sensor entrega en el pin de salida un voltaje proporcional a la corriente que mide. En la gráfica de la Figura 6 podemos observar la relación entre el voltaje de salida y la corriente sensada.

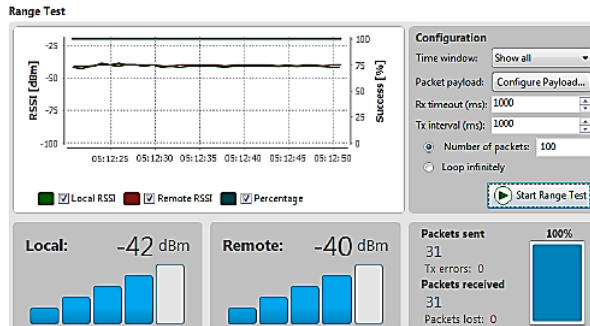


Fig. 8. Prueba de RSSI en ambiente sin obstáculos.

prueba, se reciben 31 y no hay pérdida de datos [8]. La potencia del sistema, de -42 y -40 dBm para el dispositivo local y remoto respectivamente, se encuentran en el rango de máxima potencia que puede ser recibida en una red inalámbrica.

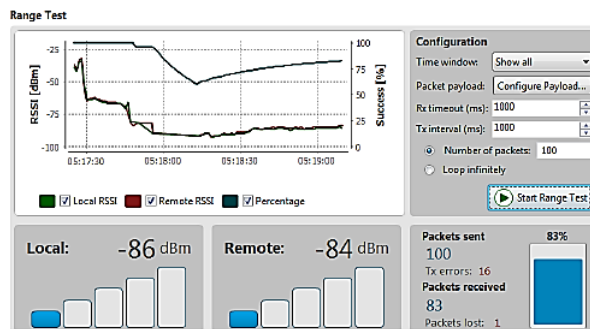


Fig. 9. Prueba de RSSI en ambiente con obstáculos.

La Fig. 9 muestra la prueba RSSI realizada en un ambiente con obstáculos (paredes de concreto, madera, entre otros). En este caso se envía el mismo número de paquetes y se usa el mismo tiempo que en la prueba anterior. El resultado muestra una pérdida de potencia, cayendo a -86 dBm y -84 dBm. Estos valores se encuentran en el rango mínimo aceptable para establecer una conexión inalámbrica.

Por otro lado, con la ayuda de un osciloscopio se verifica que la transmisión de las tramas CAN coincidan con la matriz de comunicación que se desarrolló. En las Figuras 10 y 11 se muestran algunas de las tramas CAN que se obtuvieron con el osciloscopio, la señal en color amarillo que podemos observar corresponde a la línea CAN_L del CAN Bus y su respectiva decodificación aparece dentro de las líneas azules.

Los datos que nos brinda el osciloscopio por medio de la decodificación son:

- El ID del mensaje que se está transmitiendo.
- El código de longitud de datos (DLC), los bytes de información.
- Un checksum.

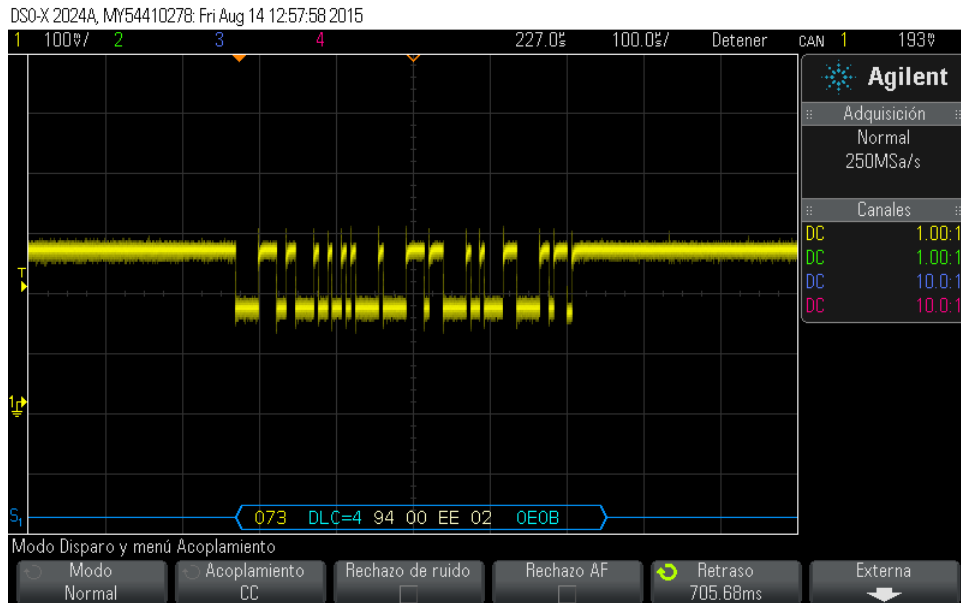


Fig. 10. Trama CAN antes.

En la Fig. 10 podemos ver que el ID del mensaje es 073, un DLC de 4 Bytes que corresponde con la información de dos sensores, la conversión analógico a digital se llevó acabo en 10 bits por lo que se necesitaron 2 bytes por sensor.

En la Fig. 11 podemos observar como a cambiado la información de los sensores pero se conserva el ID del mensaje y el DLC de esta manera comprobamos que la información es correcta.

En la Fig. 12 se muestra la red CAN–Zigbee implementada. La Fig. 12a contiene los tres nodos CAN y el nodo transmisor de Zigbee mientras que la Fig. 12b el nodo receptor de Zigbee conectado con la pantalla que posee la interfaz de usuario.

4. Conclusión

Durante las pruebas realizadas el sistema mostró el funcionamiento esperado. Las pruebas de RSSI y de transmisión de datos para Zigbee y CAN respectivamente muestran un adecuado desempeño de ambos protocolos de comunicación.

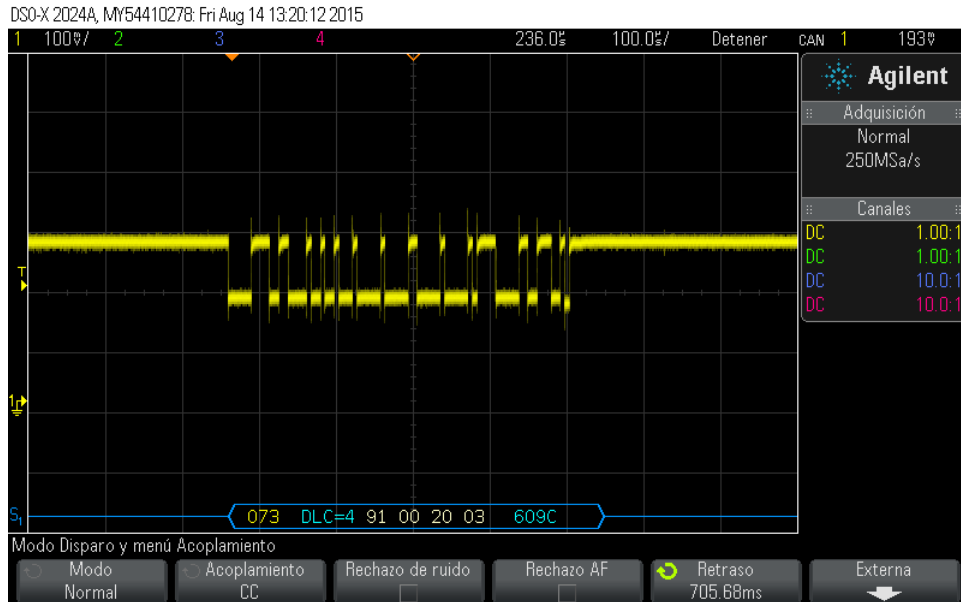
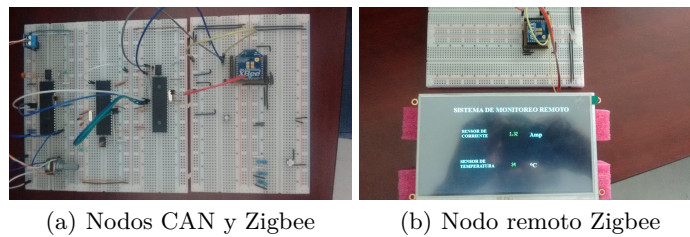


Fig. 11. Trama CAN después.



(a) Nodos CAN y Zigbee

(b) Nodo remoto Zigbee

Fig. 12. Implementación de la red CAN–Zigbee.

En el caso de Zigbee se observó que la calidad del enlace decrece en relación a la distancia entre los nodos Zigbee y los objetos que interpongan en su paso. El bus CAN al tratarse de un protocolo alámbrico no presenta el problema de Zigbee y es por esta otras ventajas que es ampliamente utilizado en sistemas que requieren cierto grado de demanda en la transmisión de datos y confiabilidad. El monitoreo de los sensores de temperatura y corriente se realiza con un muestreo cada 50 ms pantalla, mostrando así la capacidad de la red CAN para transmitir grandes cantidades de información sin pérdidas de datos.

El sistema desarrollado en este trabajo puede ser mejorado agregando más nodos a la red. Al hacer esto se podría realizar muestreo de otras variables provenientes de diversos sensores. También se podrían agregar acciones provenientes de la pantalla de la interfaz de usuario para realizar alguna tarea cuando algún sensor alcance o sobrepase algún valor de umbral preestablecido. La escalabilidad

de la red es amplia pues Zigbee soporta un numero máximo de 65000 nodos mientras que CAN hasta 30.

Referencias

1. Robert Bosch GmbH, CAN Specification Version 2.0 (1991)
2. Zigbee technology, Alliance Zigbee. Available at:<http://zigbee.org/> (Accessed on June 23, 2015).
3. Li, Jianpo and Zhu, Xuning and Tang, Ning and Sui, Jisheng: Study on ZigBee network architecture and routing algorithm. *Signal Processing Systems*.
4. Ahamed, SS Riaz: The role of zigbee technology in future data communication system. *Journal of theoretical and applied information technology*, 129–135 (2010)
5. Tindell, K.W., Hansson, H., Wellings, A.J.: Analysing real-time communications: controller area network (CAN). In: *IEEE Real-Time Systems Symposium*, pp. 259–263 (1994)
6. Oyarce, A., Aguayo, P., Martin, E.: *Guía del usuario Xbee series 2*. Ingeniería MCI Ltda (2010)
7. Ramón Pallás, A.: *Sensores y acondicionadores de señal*. Marcombo (2004)
8. Hyncica, O., Kacz, P., Fiedler, P., Bradac, Z., Kucera, P., Vrba, R.: The Zigbee experience. In: *Proceedings of the 2nd International Symposium on Communications, Control, and Signal Processing* (2006)

Sistema embebido para la detección de luz láser empleando el soft-core Nios II

Julio C. Sosa¹, Iván Domínguez-López², Adrián L. García-García²,
J.D. Oscar Barceinas-Sánchez², Anuar Jassen²

¹ Instituto Politécnico Nacional, Escuela Superior de Cómputo, México. D.F.,
México

² Instituto Politécnico Nacional, CICATA, Querétaro, Qro.
México

{jcsosa, idominguezl, agarciag, obarceinas}@ipn.mx

Resumen. Un sensor lineal CCD o CMOS es un componente esencial en muchas aplicaciones de la ingeniería y del consumo. En este trabajo se presenta la primera etapa del diseño y desarrollo de un sistema embebido para monitorear la distribución angular de la intensidad de luz láser. Se presenta la finalidad del desarrollo de ese trabajo y se muestran los primeros resultados del sistema, el cual emplea un sensor de visión lineal CCD con 3,648 píxeles y es controlado con el soft-core Nios II de Altera. Para el diseño e implementación se utilizó Quartus II software y el FPGA Cyclone IV, de Altera. Se empleó sólo el 26% de los elementos lógicos y 4 de los 132 multiplicadores embebidos (3%), del FPGA.

Palabras clave: CCD, FPGA, sistema embebido, Nios II, LLS.

Embedded System for Lazer Light Detection Using Soft-core Nios II

Abstract. A CCD or CMOS linear sensor it's an essential component in various applications of engineering and consume. In this work we will present the first stage of the design and development of an embedded system for monitoring the angular distribution of the intensity of a laser light. It is presented the purpose of developing this work and it is shown the first results of the system, which makes use of a linear vision sensor CCD with 3,648 pixels, this is controlled by the soft-core Nios II from Altera. For the design and implementation it was used the software Quartus II and the FPGA Cyclone IV both from Altera. Only the 26% of the logic elements and 4 of the 132 embedded multipliers (3%) of the FPGA were used.

Keywords: CCD, FPGA, embedded system, Nios II, LLS.

1. Introducción

Desde hace algunas décadas la importancia económica de la tribología ha tomado un gran interés. Esto es debido, por ejemplo, a que el desgaste es la principal causa de que los bienes materiales pierdan su utilidad [1].

La metodología empleada para el estudio del desgaste adhesivo, utilizando un tribómetro de perno en disco, se basa en la norma ASTM G-99-05 [2]. Esta norma establece que el resultado de la prueba se obtiene estrictamente a posteriori. Esto significa que no es posible llevar a cabo mediciones de contacto durante el desarrollo de la prueba. Al existir esta limitación es importante el desarrollar metodologías que permitan el estudio y detección del desgaste in situ y en tiempo real. Por esta razón, ha habido un gran interés en la aplicación de técnicas ópticas no destructivas, ya sea en la investigación o en la industria. En años recientes se ha utilizado la luz láser para el monitoreo de procesos de desgaste en tiempo real [3].

Para determinar el esparcimiento de luz láser o LLS (Laser Light Scattering) se emplea un esparcímetero; los elementos que lo conforman son: a) el emisor laser, b) la muestra bajo análisis, y c) el detector [4]. Bajo condiciones específicas del arreglo óptico es posible conseguir que la luz láser esparcida por la superficie caiga a lo largo de una línea recta [5], la resolución en la medición de la distribución angular de la LLS dependerá directamente del número de fotodetectores por lo que se requiere un arreglo lineal, con un gran número de fotodetectores.

En este trabajo se presenta el diseño para el control y sensado de la luz láser y una etapa de adquisición con una interfaz para la transmisión hacia un computador. Para ello se emplea: un arreglo lineal de sensores CCD, un procesador soft-core Nios II, embebido en un dispositivo lógico programable del tipo FPGA y la interfaz UART. Además de eso, se presenta la interfaz de usuario, la metodología de diseño, arquitecturas embebidas existentes para la etapa del sensado y el prototipo de diseño experimental.

2. La técnica LLS y los sistemas embebidos

2.1 Técnica de luz láser esparcida o LLS

El trabajo final consistirá en emplear la técnica LLS, la cual consiste en colocar el láser al ángulo de análisis y hacer que el detector recorra una trayectoria circular. Estos ángulos son referidos a la normal de la superficie. En una superficie idealmente lisa, Fig. 1a, todo el haz será reflejado sólo en dirección a la especular. En una superficie real, Fig. 1b, cualquier imperfección producirá el esparcimiento de luz láser, que será proporcional a los defectos de la superficie de la muestra [4].

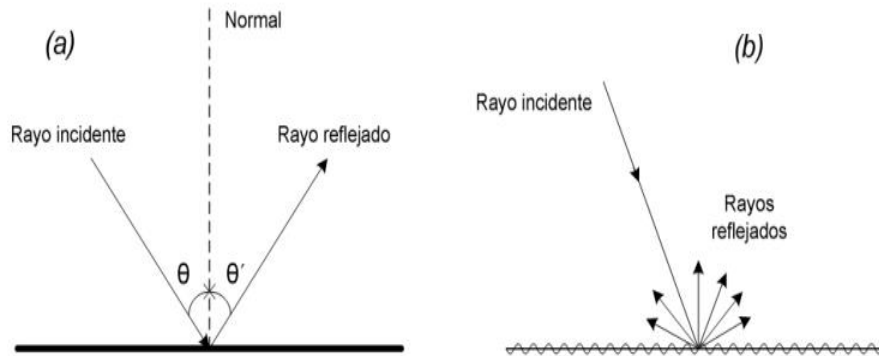


Fig. 1. Reflexión de la luz en una superficie ideal (a) y real (b).

Puesto que la luz láser se esparce en todas direcciones, después de incidir en la muestra de estudio, el detector puede ubicarse en el sitio más conveniente. Esta flexibilidad le confiere a la técnica LLS la posibilidad de adaptarse a varios sistemas tribológicos de interés para la industria [1].

Aunque en este trabajo, la técnica será utilizada para determinar o medir especificaciones de rugosidad, esta técnica también es utilizada con otros propósitos. Por ejemplo, en [6], es empleada para la medición del contenido de proteína en la leche descremada.

En [7], se utiliza para determinar especificaciones de rugosidad y para ello emplean un arreglo de fotodiodos y monitorean la distribución angular de intensidad de luz láser esparcida. La razón que exponen, del uso del arreglo de fotodiodos, es para disminuir la cantidad de datos a procesar.

En este trabajo se emplea un sensor lineal CCD con 3648 píxeles. Cada píxel tiene un grosor de $8 \mu\text{m} \times 200 \mu\text{m}$. El dispositivo es un Toshiba TCD1304AP [8].

2.2 Sistemas embebidos

Un sistema embebido es un sistema digital conformado por hardware y software y es diseñado para realizar una o varias tareas específicas [9]. En sí, los sistemas cuentan con una unidad central de procesamiento (CPU) la cual se encarga de realizar los cálculos para la tarea especificada. Una vez, con la información adquirida, por medio del sensor de visión lineal CCD y con el cálculo, se pueden accionar actuadores para realizar una tarea o bien se envían los datos adquiridos y procesados a un módulo principal que contenga y administre toda la información, como es en este caso.

Para el diseño del sistema embebido existen distintas arquitecturas, las cuales se pueden clasificar en función del tipo de dispositivo principal: las basadas en un microprocesador [10], las basadas en un microcontrolador [11], las basadas en un procesador digital de señales (DSP), las basadas en un chip hecho a la medida (SoC) y las basadas en un dispositivo lógico programable (PLD) [12].

3. Diseño de la arquitectura

3.1 Metodología de diseño

Para el diseño de la arquitectura se parte de los componentes que se van a emplear, la necesidad de un posible rediseño y la arquitectura y dispositivo a emplear. De esta manera se elige la metodología de diseño más apropiada para el sistema embebido.

Se propone que el sistema pueda alcanzar en un futuro velocidades de transmisión mayor a 12Mbps. La razón es que si bien ahora sólo se implementará la etapa de detección de luz láser, en un futuro se contempla sensar más señales o variables. Por ello se empleará para el diseño un PLD del tipo FPGA. Entonces el flujo de diseño es el mostrado en la Fig. 2, y es llamada metodología FPGA [13].

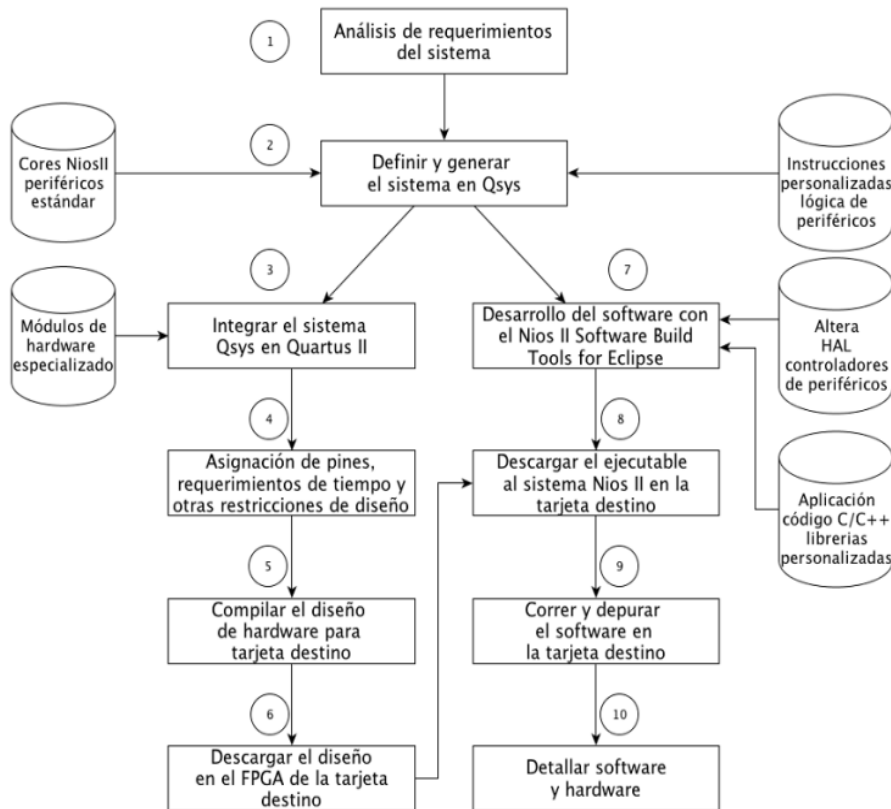


Fig. 2. Flujo de diseño de la metodología FPGA.

Se plantea utilizar un FPGA de la compañía Altera. En particular será la FPGA Cyclone IV EP4CE22F. Dicho dispositivo está montado en la tarjeta de desarrollo DE0-Nano, mostrada en la Fig. 3.



Fig. 3. Tarjeta de desarrollo DE0-Nano, de 7.5×5 .

La elección de la tarjeta, es debido a varias características, entre las que destacan: una FPGA con 22 elementos lógicos, 32 MB de memoria SDRAM, 2 KB de EEPROM, un convertidor analógico-digital (CAD) de 8 canales, de 12 bits y una dimensión de 7.5×5 .

3.2 Arquitectura propuesta

Se propone una arquitectura basada en cuatro bloques, como se muestra en la Fig. 4. El primero es una tarjeta que integra la etapa de configuración del sensor de visión lineal CCD. Un segundo módulo para la adquisición, adecuación a señal digital, procesamiento de la información y el control de la señal para la comunicación hacia la computadora. El tercer módulo, consiste en la adecuación de la señal para su transmisión vía el UART. Finalmente, se tiene un computador, donde se registrarán los datos y estos podrán ser monitorizados. Aquí se diseñará la interfaz de usuario para seleccionar el puerto COM a utilizar y sea posible mostrar en pantalla las lecturas del sensor CCD.

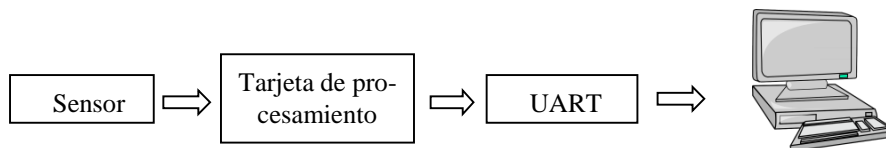


Fig. 4. Diagrama a bloques del sistema propuesto.

3.3 Etapa del sensor

En esta etapa se realiza la configuración proporcionada por el fabricante [8], como se muestra en la Fig. 5.

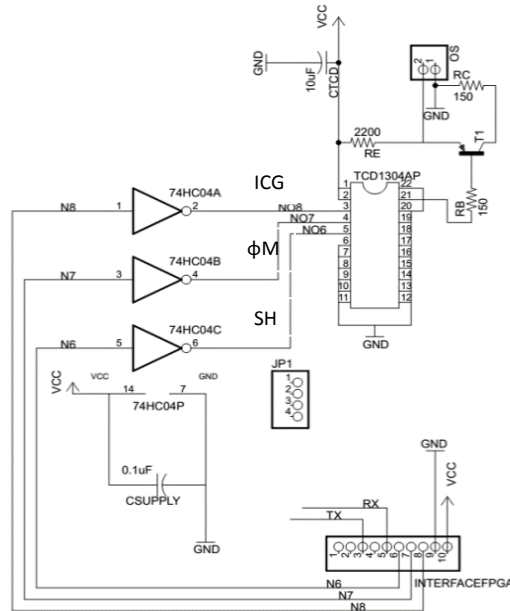


Fig. 5. Configuración del sensor de visión lineal TCD1304AP.

La configuración consiste en acondicionar las señales de control, Integration Clear Gate (ICG), Shift Gate (SH) y el Master Clock (ϕM), mediante tres inversores. Estos inversores están dentro del circuito 74HC04. Notese que debajo de los inversores están las conexiones de alimentación y tierra de dicho circuito integrado. Además de esto se recibe la salida Signal Output Voltage (OS) que es manejada por el transistor 2SA1015. La salida está en la terminal 2 del conector OS. Las señales SH, ϕM e ICG son proporcionadas y controladas por la tarjeta de procesamiento, DE0-Nano.

3.4 Tarjeta de procesamiento

Esta tarjeta tiene como componente principal la FPGA Cyclone IV EP4CE22F17C6N, de Altera; un CAD NS ADC128S022, de 8 canales y 12 bits A/D; y un banco de memoria externa SDRAM de 32 MB.

Las señal de salida del sensor CCD OS, de generación y control SH, ϕM , ICG, además de las señales de la interfaz UART RD y TD, son proporcionadas por la tarjeta DE0-Nano. Las señales, SH, ϕM e ICG, son las terminales *GPIO_2[0]*, *GPIO_2[2]* y *GPIO_2[4]*, respectivamente y la salida del sensor CCD (OS) es la terminal *Analog_In0*, todas del conector JP3. Por otro lado, las señales de comunicación RD es la *GPIO_1[5]* y TD la *GPIO_1[7]*, del conector JP1. El lenguaje de descripción de hardware utilizado para el diseño fue Verilog y la herramienta de desarrollo fue *Quartus II software, web edition* versión 13.0 con sp1.

La arquitectura para controlar estas señales, además de las necesarias para almacenar datos y transferir la información a la PC, está constituida por 6 bloques básicos, como se muestra en la Fig. 6.

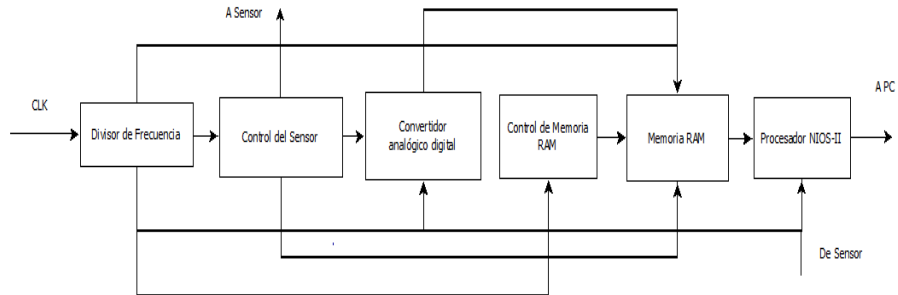


Fig. 6. Diagrama a bloques de la arquitectura en la FPGA, para el manejo y control de las señales que intervienen en el sistema.

El primer bloque, de izquierda a derecha de la Fig. 6, es un divisor de frecuencia. El segundo bloque genera las señales necesarias para el control del sensor CCD. Para el diseño del módulo se empleó el diagrama de tiempos, mostrado en la Fig. 7, en donde indica en qué momento se activa cada señal [8].

Los dos bloques siguientes, tercero y cuarto, son los que gestionan el CAD y la dirección para almacenar en una memoria RAM interna. El quinto bloque es la memoria RAM interna de 4 KB. El último bloque, es un procesador Nios II, que controla y gestiona la información de toda la arquitectura. Así mismo, funciona como un estándar de interconexión con la computadora, al cual le fue programado el protocolo UART en el procesador. Esto quiere decir que: primero al FPGA se le configura la arquitectura, todos los bloques de la Fig. 6, incluyendo el procesador Nios II. Después, el procesador Nios II es programado para que realice la gestión y el control de todo el sistema. Para su programación fue empleado el software Nios II SBT, para Eclipse [14].

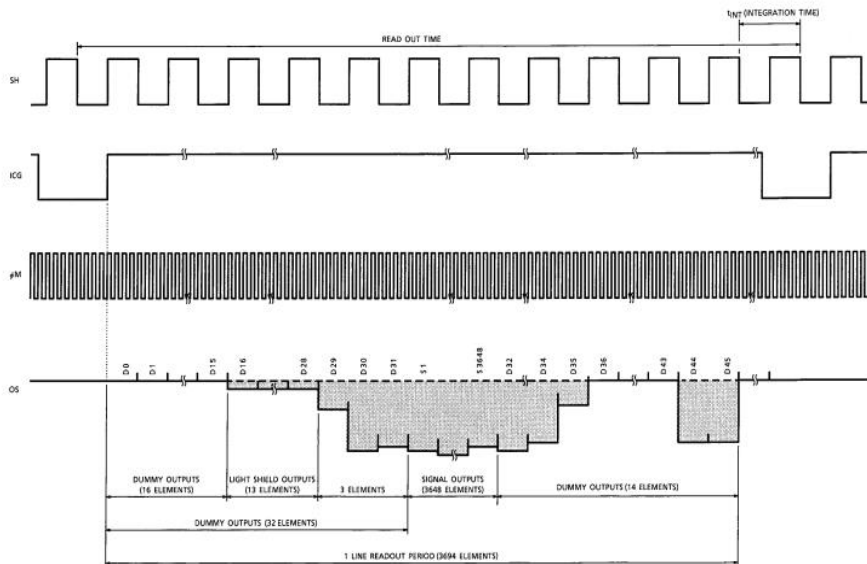


Fig. 7. Configuración del sensor de visión lineal TCD1304AP.

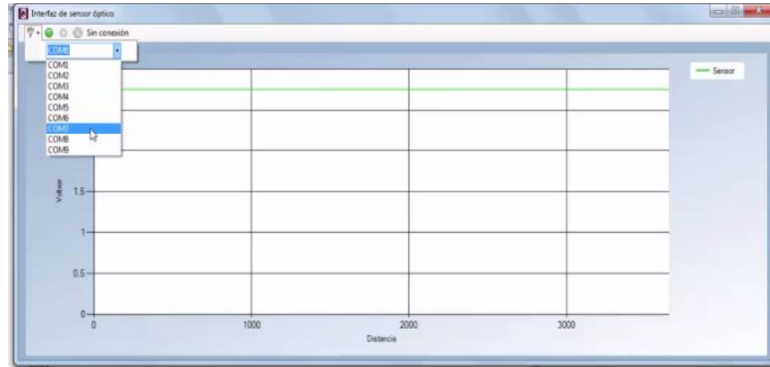


Fig. 9. Interfaz de visualización de la señal sensada por el CCD.

4. Pruebas

Debido a la sensibilidad del sensor detectada se realizaron tres pruebas, una sin iluminación (nocturna), otra de día y con iluminación artificial y una tercera que fue obstruyendo con una franja de papel al sensor de luz. Para monitorear los datos y que correspondieran con los que se grafican en la pantalla se empleó un osciloscopio, como se muestra en la Fig. 10. La primera prueba fue nocturna, con el fin de evitar la iluminación del día y de luz artificial y se empleó una videocámara para registrar la información, como se muestra en la Fig. 10 a).



Fig. 10. a) Interfaz y osciloscopio sin luz en el medio ambiente. b) Señales en la interfaz y osciloscopio, tomadas durante el día y con luz artificial.

La segunda prueba, mostrada en la Fig. 10 b), se realizó durante el día y con luz artificial, otras pruebas se realizaron con otras condiciones de iluminación y en todas se encontraba que existía una señal de error, por la iluminación del medio ambiente. Esto muestra la sensibilidad del sensor de visión lineal CCD.

Finalmente, una última prueba se muestra en la Fig. 11. En esta prueba, realizada de día y con iluminación artificial, se bloqueó la luz que incide al sensor utilizando una banda de cartón.



Fig. 11. Visualización de la interfaz con la señal del sensor CCD.

Esta prueba fue hecha así con la intención de observar el off-set que se agrega con la luz del medio ambiente. La señal se despliega en la ventana de la interfaz de la computadora.

5. Resultados y conclusiones

El diseño de la arquitectura, que controla y gestiona la información, se implementó en un FPGA; Cyclone IV de Altera. Sólo se empleó el 26% de sus elementos lógicos y el 48% de sus terminales, como indica el resultado de la simulación y síntesis de la Fig. 12.

Flow Summary	
Flow Status	Successful - Sun Jul 12 15:56:21 2015
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	DE0_NANO
Top-level Entity Name	DE0_NANO
Family	Cyclone IV E
Device	EP4CE22F17C6
Timing Models	Final
Total logic elements	5,907 / 22,320 (26 %)
Total combinational functions	5,206 / 22,320 (23 %)
Dedicated logic registers	3,858 / 22,320 (17 %)
Total registers	3927
Total pins	74 / 154 (48 %)
Total virtual pins	0
Total memory bits	196,352 / 608,256 (32 %)
Embedded Multiplier 9-bit elements	4 / 132 (3 %)
Total PLLs	1 / 4 (25 %)

Fig. 12. Resultado de la compilación y síntesis.

Se diseñaron dos circuitos PCB's de los cuales en uno se incorpora el sensor de visión lineal CCD y el otro contiene el conector DB9 y su circuito controlador MAX232. Estos PCB's se muestran en la Fig. 13. El sistema adquiere y muestra, la luz láser esparcida, a través de la interfaz de visualización. La interfaz puede almacenar, en el computador principal, la información en caso de que se quiere graficar posteriormente. La información graficada incorpora el ruido del medio ambiente.

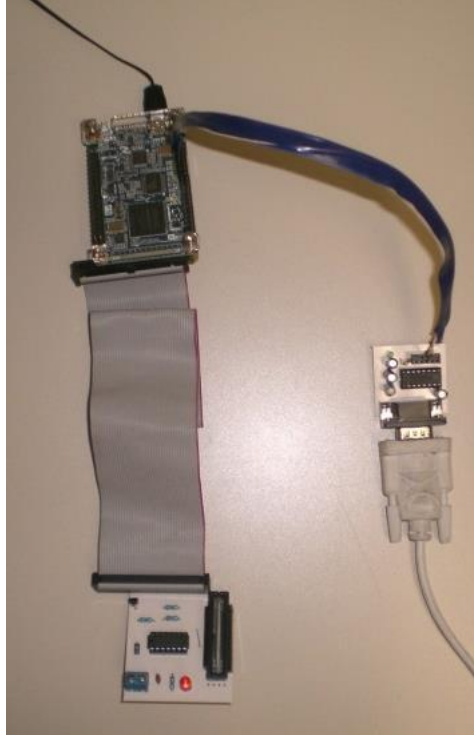


Fig. 13. PCB's del sistema embebido interconectados con la tarjeta DE0-Nano.

El sistema embebido diseñado detecta la luz láser esparcida analógica y es convertida y procesada en la tarjeta de desarrollo DE0-Nano. Posteriormente es transmitida mediante el puerto UART hacía la computadora en donde es graficada la señal. Se presenta el inconveniente de que la sensibilidad de sensor de visión lineal CCD es tal que al funcionar durante el día o con iluminación artificial existe una señal incorporada a la señal de la luz láser esparcida. Por tal razón se propone agregar o incorporar un filtro para separar la luz del ambiente y leer sólo la luz láser esparcida.

Agradecimientos. Este trabajo ha sido financiado por la Secretaria de Investigación y Posgrado del IPN, con el proyecto de número de registro SIP: 20151013.

Referencias

1. Domínguez, I., Huerta, J.A., Montes, R.I.: Mediciones de cambio de intensidad en luz laser esparcida, aplicada al monitoreo de desgaste. Simposio de metrología, pp. 1-5 (2006)
2. ASTM G 99-05: Standard Test Method for Wear Testing with a Pin-on-Disc Apparatus. ASTM International. www.astm.org (2015)
3. Martínez-Fuentes, V., Domínguez-López, I., García-García, A.L.: Surface texture changes followed-up in real time during the initial wear transient of dry sliding of

- steel against several metals using laser light scattering, *Wear*, Vol. 271, pp. 994–998 (2011)
4. Moreno, A., Miramontes, G., Sifuentes, C., García, E., Huerta, J.A.: Una Herramienta Computacional para Simulación del Esparcimiento de Luz Láser en Superficies Rugosas, *Revista Colombiana de Física*, Vol. 43, No. 1 (2011)
 5. Martínez Fuentes, V. Domínguez López, I., García García A.L.: Estudio teórico-experimental del esparcimiento de luz láser por superficies sometidas a desgaste, p.p. 371 (1–6), CIINDET (2008)
 6. Yong, Q., Zhen, Z.: Study of Measurement of Protein Content Based on Laser Light Scattering Technology. International Conference on Control, Automation and Systems Engineering (CASE), pp. 1–4 (Julio 2011)
 7. Ortiz, J.D.: Desarrollo y Aplicación de Instrumentación para la Medición de Esparcimiento de Luz Láser. Instituto Politecnico Nacional, CICATA-QRO, Tesis Doctoral, (Junio 2010)
 8. Toshiba: Data Sheet: CCD Linear Image Sensor TCD1304AP (2001)
 9. Galeano, G.: Programación de sistemas embebidos en C. Alfaomega Grupo Editor, S.A. de C.V. (2011)
 10. <https://www.raspberrypi.org/>
 11. Angulo, J.M., Angulo, I., Etxebarria, A.: Microcontroladores PIC: Diseño práctico de aplicaciones. Mc. Graw Hill (2007)
 12. Sosa, J.C., Martínez, Y., García, V., Hernández, R.: Estudio y elección de un softcore para la enseñanza de sistemas embebidos. 24a. Reunión internacional de otoño de comunicaciones, computación, electrónica, automatización, robótica y exposición industrial, ROC&C. Ponencia 50 (2013)
 13. Gajski, D.D., Abdi, S., Gerstlauer, A., Schirner, G.: *Embedded System Design: Modeling, Synthesis and Verification*, Springer (2009)
 14. Altera: *Nios II Software Developer's Handbook: Getting Started with the Graphical User Interface* (January 2014)
 15. Texas Instruments: *MAX232x Dual EIA-232 Driver/Receivers* (2014)

Solución al problema de ordenación usando PipeLine y Binary Tree como composiciones paralelas de alto nivel

Mario Rossainz-López, Ivo H. Pineda-Torres, Patricia Domínguez Mirón

Benemérita Universidad Autónoma de Puebla,
México

{rossainz, ipineda}@cs.buap.mx, paty.dguez.m@gmail.com

Abstract. This paper proposes the model of the High Level Parallel Compositions or CPANs to communication patterns / interaction Pipeline and TreeDV for implementing a sorting algorithm by Structured Parallel Programming approach based on the concept of Parallel Objects. The CPANs TreeDV and PipeLine are displayed using the paradigm of object orientation and sorting problem is solved using two different algorithms; it is using a pipeline process to sort a dataset in disordered (CPAN Pipe) and one that by quick sort uses a binary tree for the ordering of the same dataset disordered by divide and conquer technique (CPAN TreeDVQS). Each proposal of CPAN contains a predefined set of restrictions of synchronization between processes (maximum parallelism, mutual exclusion and synchronization of producer-consumer type), and the use of synchronous, asynchronous and asynchronous future communication modes. Sorting algorithms, their design and implementation as CPANs and comparative performance metrics on a parallel machine 64 processors are shown.

Keywords: CPAN, programación paralela estructurada, objetos paralelos, programación orientada a objetos, pipeline, binary tree, divide y vencerás.

1. Introducción

Actualmente dentro del ámbito de la programación paralela uno de los problemas abiertos de mayor interés es el de la falta de aceptación de entornos de programación paralela estructurada para desarrollar aplicaciones paralelas por parte de los usuarios. Actualmente existen trabajos previos a esta investigación donde se hace patente la necesidad de contar con patrones paralelos de comunicación [1]. En la literatura existen varias propuestas, pero todas ellas coinciden en la importancia de determinar un conjunto completo de patrones y tratar de definir una semántica para ellos [6]. La tendencia para esto es en la actualidad el uso de enfoques de programación orientados a objetos, pues se ha visto que el definir objetos paralelos para el desarrollo de nuevas propuestas de metodologías, modelos y patrones de comunicación de programación paralela ha dado buenos resultados [4, 8]. Las Composiciones Paralelas de Alto Nivel o CPANs que aquí se proponen son patrones paralelos de comunicación bien

definidos y lógicamente estructurados que, una vez identificados en términos de sus componentes y de su esquema de comunicación, pueden llevarse a la práctica como constructos añadidos a un lenguaje de programación orientado a objetos y estar disponibles como abstracciones de alto nivel en las aplicaciones del usuario [11, 13]. Esto es posible gracias a la adopción del modelo de los Objetos Paralelos y con ellos construir el PipeLine y el BinaryTree como composición de éstos en alto nivel y general dichas estructuras de interconexión de procesos para resolver el problema de la ordenación.

2. Las composiciones paralelas de alto nivel o CPANs

Un CPAN es la composición de un conjunto de objetos paralelos de tres tipos: Un objeto manager que representa al CPAN en sí mismo y hace de él una abstracción encapsulada que oculta su estructura interna. El manager controla las referencias de un conjunto de objetos (un objeto denominado Colector y varios objetos denominados Stage), que representan los componentes del CPAN y cuya ejecución se lleva a cabo en paralelo y debe ser coordinada por el propio manager. Los objetos Stage son los encargados de encapsular una interfaz tipo cliente-servidor que se establece entre el manager y los objetos esclavos (objetos pasivos que contienen el algoritmo secuencial de la solución de un problema). Y un objeto Colector que es un objeto encargado de almacenar en paralelo los resultados que le lleguen de los objetos stage que tenga conectados, ver Fig. 1.

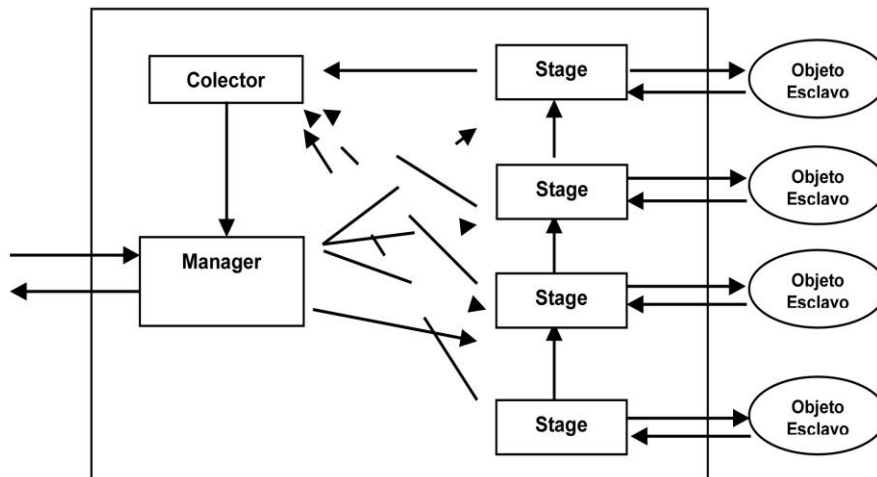


Fig. 1. Estructura Interna de un CPAN.

Los objetos manager, colector y stages son Objetos Paralelos (PO) [5, 6]. Los Objetos Paralelos tienen capacidad de ejecución en sí mismos [5]. Las aplicaciones dentro del modelo PO pueden explotar tanto el paralelismo entre objetos (inter-object) como el paralelismo interno de ellos (intra-object). Un PO tiene una estructura similar

a la de un objeto en Smalltalk, pero además incluye una política de planificación determinada a priori que especifica la forma de sincronizar una o más operaciones de la clase del objeto susceptibles de invocarse en paralelo [5, 7]. Los objetos paralelos soportan herencia múltiple, que permite derivar una nueva especificación de PO completa a partir de una que ya existe. Un CPAN cuenta con las siguientes propiedades que se pueden estudiar a detalle en [15]: Modo asíncrono y futuro asíncrono de comunicación entre los objetos paralelos del CPAN [6], objetos con paralelismo interno., disponibilidad de mecanismos de sincronización; Paralelismo Máximo (MaxPar), Exclusión Mutua (MuTex) y Sincronización (Sync) del tipo Productor-Consumidor, disponibilidad de control de tipos genéricos, transparencia en la distribución de aplicaciones paralela y rendimiento satisfactorio: Programabilidad-Portabilidad-Performance.

2.1 Clases base que conforman los Objetos Paralelos de un CPAN

En los PO las clases básicas necesarias para definir los objetos manager, colector, stages de un CPAN son: Una instancia de una clase concreta derivada de la clase ComponentManager (llamada manager) representa un CPAN dentro de una aplicación programada según el modelo de objeto paralelo. Las instancias (llamadas stages) de una clase concreta derivada de la clase ComponentStage se conectan entre sí para implementar una composición de stages. Cada stage ordena la ejecución en paralelo de un objeto llamado esclavo (slave), que es controlado por el propio stage. Por otro lado, es preciso señalar que la creación de los stages y de los colectores y sus interacciones posteriores son manejadas transparentemente al código de la aplicación por el manager. Desde el punto de vista de un usuario interesado en reutilizar el comportamiento paralelo ya definido en algunas clases CPAN, la clase de interés será la del manager. Cuando un usuario está interesado en usar un CPAN dentro de una aplicación, tiene que crear una instancia de una clase manager concreta, esto es, una que implemente el comportamiento paralelo requerido por la aplicación y que la inicialice con la referencia apropiada a los objetos esclavos que van a ser controlados por cada stage, así como indicar el nombre del método solicitado, para más detalles ver [12].

3. El CPAN pipeline

La técnica del procesamiento paralelo del pipeline se presenta como una Composición Paralela de Alto Nivel aplicable a la solución de un amplio rango de problemas que son parcialmente secuenciales en su naturaleza, de tal forma que el CPAN Pipe garantiza la paralelización de código del algoritmo secuencial.

3.1 La técnica del pipeline

La idea es dividir un problema a resolver en una serie de tareas que tienen que ser completadas una después de otra. En un pipeline cada tarea puede ser ejecutada por un proceso, un thread o un procesador independiente [9], ver Fig. 2.

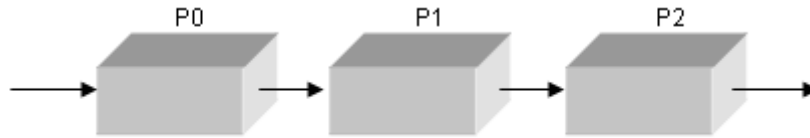


Fig. 2. Un Pipeline.

Algunas veces a los procesos del pipeline se les llama etapas (stages) del pipeline [10]. Cada etapa puede contribuir a la solución del problema total y puede pasar la información necesaria a la siguiente etapa del pipeline. Este tipo de paralelismo es visto muchas veces como una forma de descomposición funcional, ya que el problema es dividido en funciones separadas que pueden ser ejecutadas individual e independientemente; pero con esta técnica las funciones se ejecutan en sucesión. [9].

3.2 El modelo del pipeline como un CPAN

La Fig.3 representa el patrón paralelo de comunicación PipeLine como un CPAN.

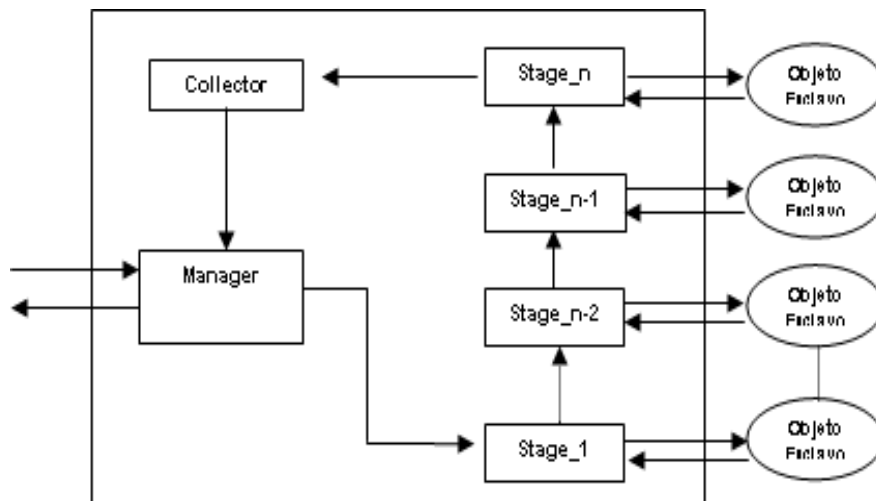


Fig. 3. El Pipeline como un CPAN.

Los objetos Manager y stage_i del modelo gráfico del CpanPipe son instancias de clases concretas, digamos PipeManager y PipeStage, que heredan las características de las clases base ComponentManager y ComponentStage, respectivamente, para poder así redefinir los métodos abstractos de las superclases. El objeto Collector es el único que será una instancia de la clase base ComponentCollector ya que ésta es una clase concreta. Una vez creados los objetos, y conectados de manera apropiada de acuerdo al patrón paralelo Pipeline, se tendrá entonces un CPAN para un tipo de patrón paralelo específico, después ya se puede resolver la asignación de los objetos esclavos asociados a los stages. Para ver más detalles consultar [12].

3.3 El algoritmo de ordenación usando un pipeline

En el funcionamiento de un algoritmo paralelo de ordenación basado en el pipeline se distinguen 3 fases según se menciona en [2, 3]:

- La carga inicial: Se pretende hacer llegar datos a todos los procesos asociados a las etapas del pipeline. En esta fase los procesos suelen ejecutar el mismo código que en la segunda fase, la diferencia consiste en que hay que inicializarlos de forma adecuada para que reciban el primer dato, que les va a llegar de la etapa anterior o de la carga inicial del programa.
- El procesamiento de la secuencia de datos con la máxima eficiencia: Los procesos se comportan de manera cíclica en su ejecución. Admiten datos de la etapa anterior, los procesan y envían el resultado a la etapa siguiente. Cada proceso tiene que estar sincronizado con el de la etapa anterior para que no le envíe nuevos datos cuando todavía no ha terminado de procesar los datos corrientes; pero también con el de la etapa siguiente, para no enviar el resultado hasta que el proceso de dicha etapa no esté preparado para recibirlo. El último proceso tiene un comportamiento especial con respecto a los procesos asociados a las etapas anteriores ya que tiene que ejecutar una rutina o código de salida y presentación de resultados. Su funcionamiento consiste en obtener los datos que le envía el proceso de la última etapa del pipeline y enviarlos a un dispositivo de salida o enviar una condición de terminación al programa principal. La serie de resultados que produce el último proceso habrá de coincidir con el resultado esperado del algoritmo que se ha paralelizado, si el pipeline ha sido paralelizado correctamente.
- La descarga: En esta última etapa los procesos envían el resultado del último dato procesado y ellos mismos detectan la situación de terminación, pues ya no van a recibir más datos de la secuencia de entrada y no se debe suponer ningún control global en el programa que les indique cuando tienen que terminar. Para que los procesos transmitan el dato almacenado en sus etapas antes de terminar, se suele introducir un valor especial al final de la secuencia de entrada que sirve para descargar el pipeline.

La implementación del algoritmo paralelo de ordenación consiste en un pipeline de procesos que es alimentado con una secuencia desordenada de números enteros por una rutina o código de entrada. El resultado obtenido como salida es la secuencia de enteros ordenada de menor a mayor. El número de valores de la secuencia de entrada no puede ser mayor que el número de etapas del pipeline. Cada uno de los procesos del pipeline tiene capacidad para almacenar un número entero, que será el mayor que haya recibido hasta ese momento de la etapa anterior. En cada iteración, un proceso recibe un entero, lo compara con el que tenía almacenado y envía el menor de ambos a la siguiente etapa del pipeline, mientras que el mayor es almacenado [2], [3]. Éste esquema representa el algoritmo paralelo de ordenación utilizando un Pipeline el cual ha sido implementado como CpanPipe

4. El CPAN treeDV

Se presenta la técnica de programación Divide y Vencerás como un CPAN haciendo uso de un patrón de comunicación en forma de árbol binario, aplicable a un amplio rango de problemas que pueden ser paralelizables bajo este esquema.

4.1 La técnica de divide y vencerás usando un árbol binario

Consideremos el algoritmo de divide y vencerás paralelo representado a través de un árbol binario (Fig.4.), donde cada nodo es un proceso.

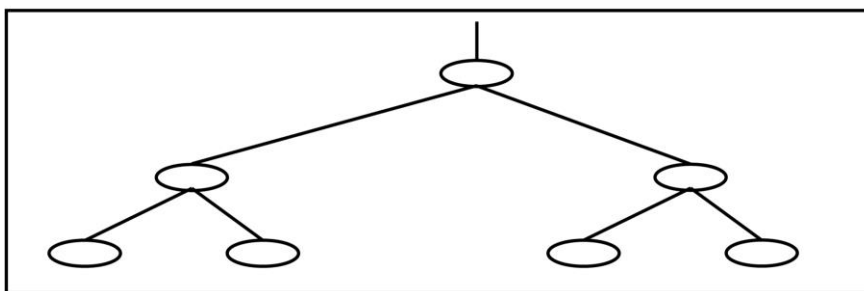


Fig. 4. Representación Gráfica de un árbol binario.

El nodo raíz del árbol tiene como entrada un problema completo, que se divide en dos partes y envía una parte al nodo hijo izquierdo y la otra parte al nodo hijo derecho. Este proceso de división es repetido en los niveles más altos del árbol. Eventualmente cualquier nodo hoja tiene como entrada un problema dado por su nodo padre, lo resuelve y la solución (que es la salida del nodo hoja) es enviada a su progenitor. Cualquier nodo padre en el árbol obtendrá dos soluciones parciales de sus hijos y las combinará para dar como salida una simple solución que será la salida del nodo padre. Finalmente el nodo raíz dará como salida la solución completa del problema.

4.2 La técnica de divide y vencerás como un CPAN

La representación del patrón treeDV que define la técnica de Divide y Vencerás como CPAN tiene su modelo representado en la fig. 5. A diferencia del modelo anterior, donde los objetos esclavos eran predeterminados, sólo un objeto esclavo es predefinido estáticamente y asociado al primer stage del árbol en este modelo, los siguientes objetos esclavos serán creados internamente por los propios stages de forma dinámica, pues los niveles del árbol dependen del problema a resolver y no se conoce a priori el número de nodos que pueda tener el árbol, ni tampoco su nivel de profundidad. No obstante, al igual que en el modelo anterior, los objetos Manager y stage_i de la fig. 5, son instancias de clases concretas, digamos TreeDVManager y TreeDVStage, que heredan de las clases base denominadas ComponenManager y ComponentStage, respectivamente. El objeto Collector es aquí también una instancia

de la clase base ComponentCollector. De acuerdo con la técnica de Divide y Vencerás se crea la clase TreeDVManager, que hereda de ComponentManager e implementa un patrón de comunicación de un árbol binario. Los nodos del árbol binario están representados por los stages que son objetos de la clase TreeDVStage que, a su vez, hereda de ComponentStage. Cualquier instancia de la clase TreeDVManager se encarga sólo del primer stage o nodo raíz del árbol en el momento de su inicialización. Durante la ejecución de una petición de servicio, la raíz del árbol binario, es decir, el primer stage de la estructura, es iniciada por el manager; además, todo nodo padre creado en los niveles más bajos del árbol ordenarán internamente la ejecución de sus respectivos nodos hijos, dentro del proceso de obtención de la solución del problema en esta técnica de diseño. Para ver más detalles consultar [12].

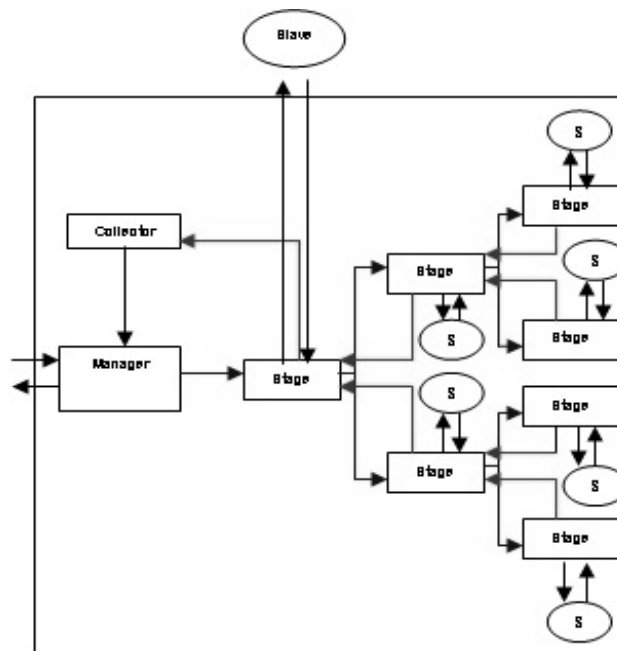


Fig. 5. El CPAN de un TreeDV.

4.3 El algoritmo de ordenación QuickSort usando el CPAN treeDV

El problema de ordenación rápida o Quicksort fue creado por Hoare y se basa en el paradigma de divide y vencerás. Como primer paso el algoritmo selecciona como pivote uno de los elementos del conjunto de datos que haya que ordenar. A continuación el conjunto se parte a ambos lados del pivote. Se desplazan los elementos de tal manera que los que sean mayores que el pivote queden a su derecha, mientras que los que sean menores queden a su izquierda. Posteriormente las partes del conjunto que quedan a ambos lados del pivote se ordenan independientemente de forma paralela y recursiva, en este caso a través de los objetos stage del modelo del CPAN TreeDV. El resultado final es un conjunto completamente ordenado.

5. Rendimiento

El análisis de rendimiento del CpanPipe y del CpanTreeDV para la implementación de los algoritmos paralelos de ordenamiento se llevó a cabo en una computadora paralela con 64 procesadores, 8 Gb de memoria principal, buses de alta velocidad y arquitectura de memoria compartida distribuida. Las medidas de rendimiento obtenidas en la ejecución de los CPANs que resuelven el problema de ordenación se llevó a cabo con las siguientes condiciones de ejecución

- Implementación paralela del algoritmo secuencial de ordenación basado en un pipeline para el caso del Cpan Pipe e implementación paralela del algoritmo secuencial de ordenación QuickSort basado en un árbol binario utilizando la técnica de Divide y Vencerás para el caso del CpanTreeDV
- En ambos casos, tanto para el CPan Pipe como para el Cpan TreeDV, se implementó el mismo algoritmo secuencial de comparación de valores en cada uno de los objetos esclavos asociados a las etapas o stage de los CPANs
- La ordenación de un conjunto de 50000 números enteros obtenidos de manera aleatoria en un rango de 0 a 50000, lo que ha permitido suponer una carga suficiente para los procesadores y con ello apreciar la mejora del rendimiento de los Cpan Pipe y TreeDV
- Ejecución del CpanPipe y del CpanTreeDV para 2, 4, 8, 16 y 32 procesadores con dedicación exclusiva para resolver el problema de ordenación en ambos casos.

Las tablas 1 y 2 y las Figuras 6 y 7 muestran las series de medidas obtenidas en los CPANs mencionados incluyendo las de sus correspondientes versiones secuenciales, tiempo de ejecución en segundos, ciclos ejecutados por instrucción, la magnitud del speedup encontrado y la cota superior de la magnitud del speedup utilizando para ello la ley de Amdahl.

Tabla 1. Rendimiento del Cpan Pipe Paralelo para la ordenación por de 50000 números enteros.

CPAN Pipe	Pipe Secuencial	cpuset2	cpuset4	cpuset8	cpuset16	cpuset32
Tiempo de ejecución en seg.	238.50	127.27	123.83	116.97	115.63	111.03
Consumo de tiempo en seg. de CPU	231.82	224.79	217.80	203.98	198.30	191.89
CPI	1.862	0.909	0.904	0.901	0.886	0.871
Speedup	1.00	1.87	1.93	2.04	2.06	2.15
Amdalh	1.00	1.89	3.39	5.63	8.42	11.19

Tabla 2. Rendimiento del TreeDV Paralelo para la ordenación por de 50000 números enteros.

CPAN TreeDVQS	TreeDVQS Secuencial	cpuset2	cpuset4	Cpuset8	cpuset16	cpu set32
Tiempo de ejecución en seg.	20.55	11.48	7.21	4.59	3.68	3.40
Consumo de tiempo en seg. de CPU	7.60	7.18	6.31	6.88	6.45	6.35
CPI	1.261	0.900	0.892	0.858	0.849	0.836
Sppedup	1.00	1.79	2.85	4.48	5.58	6.04
Amdalh	1.00	1.82	3.08	4.71	6.40	7.80

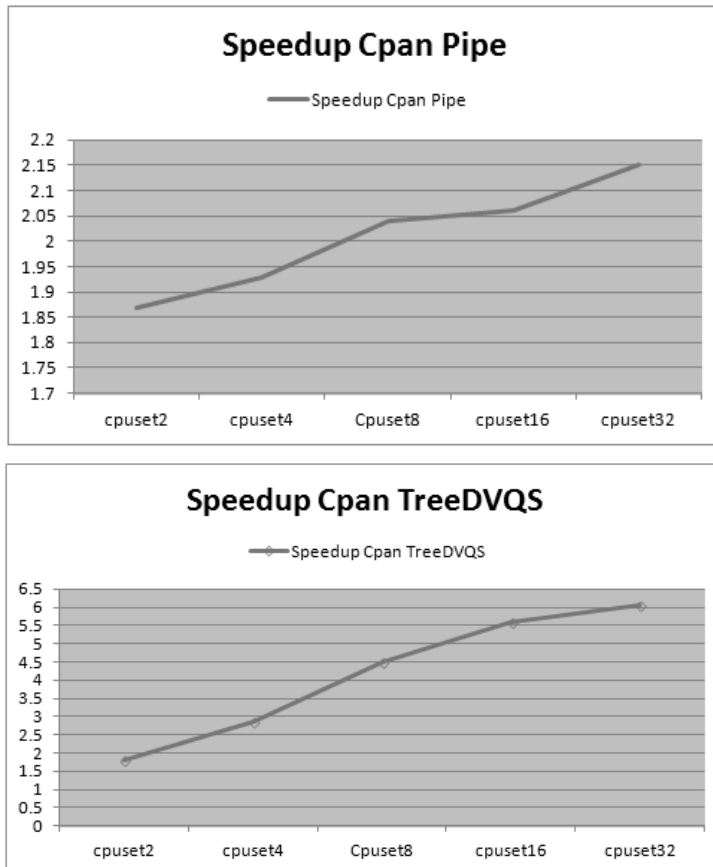


Fig. 6. Escalabilidad de la magnitud del Speedup encontrado para el Cpan Pipe y Cpan TreeDV con 2, 4, 8, 16 y 32 procesadores exclusivos.

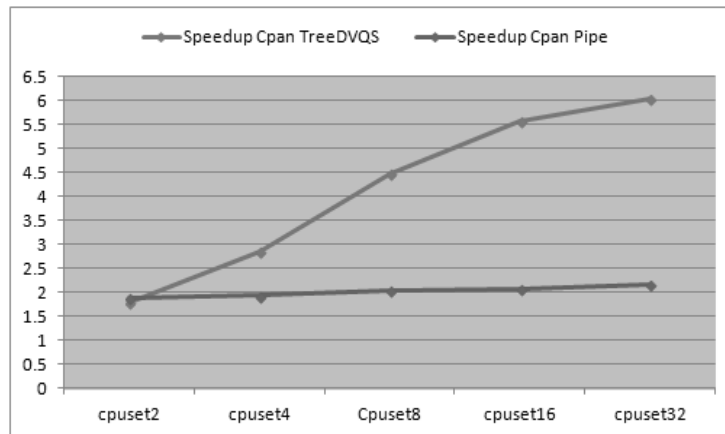


Fig. 7. Comparativo de la magnitud del Speedup de los Cpan Pipe y TreeDV en la solución del problema de ordenación con 2, 4, 8, 16 y 32 procesadores exclusivos.

6. Conclusiones

Se han implementado los CPAN PipeLine y TreeDV los cuales forman parte de una librería de clases de Objetos Paralelos que proporciona al usuario los patrones citados como Composiciones Paralelas de Alto Nivel más aparte otros patrones de comunicación como las granjas o farms y los árboles o Tree y cuyos detalles están publicados en [14]. Los CPANs Pipe y TreeDV pueden ser explotados gracias a la adopción del enfoque orientado a objetos para definir nuevos patrones en base a los ya construidos [14]. Se han transformado algoritmos secuenciales de ordenación en algoritmos paralelizables mediante el uso de un Pipeline y de un árbol binario utilizando el paradigma de Divide y Vencerás como CPANs. Se ha obtenido soluciones elegantes al problema de ordenación usando mediante dos tipos de algoritmos; aquél que utiliza descomposición funcional en forma de cauce y el de ordenación rápida o QuickSort. Se han programado las restricciones de sincronización sugeridas por el modelo del CPAN: el paralelismo máximo (MaxPar), la exclusión mutua (Mutex) y la sincronización de comunicación de procesos productor/consumidor (Sync) en ambos CPANs.

Se ha probado el rendimiento de los CPANs Pipe y TreeDV mediante métricas de Speedup, Ley de Amdahl y eficiencia para demostrar que el comportamiento paralelo de los CPANs propuestos es mejor que su contraparte secuencial.

Referencias

1. Bacci, D., Pelagatti, V.: SkIE: A Heterogeneous Environment for HPC Applications. *Parallel Computing* 25, pp. 1827–52 (1999)
2. Barry, W., Allen, M.: *Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers*. Prentice Hall (1999)

3. Blelloch, Guy E: Programming Parallel Algorithms. Communications of the ACM, Vol. 39, No. 3 (1996)
4. Brinch, H.: Model Programs for Computational Science: A programming methodology for multicomputers. Concurrency: Practice and Experience, Vol. 5, No. 5, pp. 407–423 (1993)
5. Corradi, A., Leonardi, L.: PO Constraints as tools to synchronize active objects. Journal Object Oriented Programming 10, pp. 42–53 (1991)
6. Corradi, A.L., Zambonelli, F.: Experiences toward an Object-Oriented Approach to Structured Parallel Programming. DEIS technical report no. DEIS-LIA-95-007 (1995)
7. Danelutto, M., Orlando, S.: Parallel Programming Models Based on Restricted Computation Structure Approach. Technical Report, Dpt. Informatica, Università de Pisa (1999)
8. Darlington: Parallel Programming Using Skeleton Functions. In: Proceedings PARLE'93, Munich (1993)
9. Robbins, K.A., Robbins, S.: UNIX Programación Práctica. Guía para la concurrencia, la comunicación y los multihilos. Prentice Hall (1999)
10. Roosta, Séller: Parallel Processing and Parallel Algorithms. Theory and Computation, Springer (1999)
11. Rossainz, M.: Una Metodología de Programación Basada en Composiciones Paralelas de Alto Nivel (CPANs). Universidad de Granada (2005)
12. Rossainz M., Capel M.: An Approach to Structured Parallel Programming Based on a Composition of Parallel Objects. In: Congreso Español de Informática CEDI-2005, XVI Jornadas de Paralelismo, Editorial Thomson, España (2005)
13. Rossainz, M., Capel, M.: A Parallel Programming Methodology using Communication Patterns named CPANS or Composition of Parallel Object. In: Proceedings of 20th European Modeling & Simulation Symposium, Campora S. Giovanni (ed.), Italy (2008)
14. Rossainz, M., Capel, M.: Compositions of Parallel Object to Implement Communication Patterns. In: Proceedings of XXIII Jornadas de Paralelismo, pp. 8–13, Elche, Spain (2012)
15. Rossainz, M., Pineda, I., Dominguez, P.: Análisis y Definición del Modelo de las Composiciones Paralelas de Alto Nivel llamadas CPANs. Modelos Matemáticos y TIC: Teoría y Aplicaciones. Dirección de Fomento Editorial, pp. 1–19, México (2014)

Un sistema experto difuso en la Web para diagnóstico de diabetes

Viridiana Cruz-Gutiérrez, Abraham Sánchez-López

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

`viricruz@rockkruz.net, asanchez@cs.buap.mx`

Resumen. La Inteligencia Artificial se ha proyectado a múltiples aspectos de la medicina, como la investigación, la asistencia clínica y la gestión sanitaria. Es por ello que en este artículo se propone un Sistema Experto Difuso para diagnóstico de diabetes mellitus usando la librería `jFuzzyLogic`, que ofrece la implementación de inferencia difusa y el API de Java para Servicios Web XML (JAX-WS). También, se describe el diseño de la base del conocimiento mediante técnicas de Ingeniería del Conocimiento e Inteligencia Artificial. El conocimiento plasmado contiene la información y experiencia de un especialista en medicina familiar del Instituto Mexicano del Seguro Social (IMSS). Con el procesamiento y uso de este conocimiento es posible diagnosticar distintos tipos de diabetes, tomando como base a los síntomas del paciente. Al final de este trabajo se presentan los resultados preliminares de la implementación.

Palabras clave: Base de conocimiento, diabetes mellitus, sistema experto.

A Web-based Fuzzy Expert System for Diabetes Diagnostics

Abstract. Artificial Intelligence is projected to multiple medicine aspects, including research, clinical care and health management. That is why this article proposes a Fuzzy Expert System for diagnosis of diabetes mellitus using the library `jFuzzyLogic` which offers implementation of fuzzy inference and Java API for XML Web Services (JAX-WS). Additionally, the design of the knowledge base is described by Knowledge Engineering techniques and Artificial Intelligence. The knowledge embodied in the system, contains the information and expertise of a specialist in family medicine at the Mexican Institute of Social Security (IMSS). With the processing and use of this knowledge is possible to diagnose different kinds of diabetes, based on the patient's symptoms. At the end of this paper the preliminary results of the implementation are presented.

Keywords: Knowledge base, diabetes mellitus, expert system.

1. Introducción

La medicina ha sido una disciplina que se ha apoyado en el uso de tecnologías y software. Surge así, la informática médica, la cual se relaciona con los sistemas de la información y de comunicación, los lenguajes médicos formales, las guías de práctica clínica, la Inteligencia Artificial y la cibernética [1].

Para dar un diagnóstico certero de diabetes mellitus, el médico se basa en los síntomas del paciente y en los criterios bioquímicos con o sin síntomas clínicos [2]. La poca experiencia de médicos principiantes, puede provocar que se realice un diagnóstico equívoco y que las recomendaciones no sean las adecuadas, poniendo en riesgo al paciente; por ello en esta investigación se muestra una propuesta de un Sistema Experto Difuso (SED) para apoyo a los médicos en el diagnóstico y prevención de diabetes, cuya base de conocimiento se diseñó mediante las técnicas de la Ingeniería del Conocimiento y que será posible consultar desde una Aplicación Web, para reducir el tiempo que les toma a los médicos dar un diagnóstico y para que complementen los consejos generales de cuidado con otros más específicos.

En la *Sección 2* se presenta una breve reseña de los conceptos más importantes acerca de la lógica difusa, la ingeniería del conocimiento y la representación del conocimiento, después, en la *Sección 3* se describen los pasos que se siguieron para el diseño de la base del conocimiento y la implementación en la Web, para que posteriormente en la *Sección 4* se presenten las pruebas de la propuesta, y los resultados obtenidos, finalmente en la *Sección 5* se dan a conocer las conclusiones y el trabajo futuro.

2. Marco teórico

Los Sistemas Expertos (SE) pertenecen a una de las áreas de mayor éxito en el campo de la Inteligencia Artificial; también son conocidos como sistemas expertos basados en conocimiento [4]. En los años setenta, en la Universidad de Stanford se desarrolló MYCIN que apoyaba a los médicos en la investigación y diagnóstico de enfermedades infecciosas de la sangre.

El conocimiento de los SE puede obtenerse por experiencia de un experto humano y/o consulta de conocimientos que están disponibles en bibliografía especializada; éste se puede documentar, revisar, complementar y utilizar en diferentes lugares y tiempos, permitiendo mejorar la toma de decisiones [5].

2.1. Lógica difusa

A diferencia de la lógica clásica, la lógica difusa [6], no tiene bien definidos los umbrales de decisión, por lo que proporciona un medio para enfrentar situaciones del mundo real que requieren razonamiento aproximado para manipular información cualitativa más que cuantitativa y situaciones complejas y dinámicas, caracterizadas en el lenguaje natural.

La lógica difusa se utiliza comúnmente para procesos difíciles de modelar matemáticamente, ya que el diseño y sintonización del controlador difuso se basa únicamente en la experiencia del experto en el proceso.

Los conceptos importantes en la lógica difusa son [8]:

- *Variable lingüística*: Concepto a calificar de forma difusa, por ejemplo: edad, altura, temperatura, etc.
- *Universo de discurso*: Rango de valores que pueden tomar los elementos que poseen la propiedad expresada por la variable lingüística.
- *Valor lingüístico*: Diferentes clasificaciones que se efectúan sobre la variable lingüística. En el caso de la temperatura, se puede dividir el universo de discurso en los diferentes valores lingüísticos: frío, templado, caliente.
- *Función de pertenencia*: Aplicación que se asocia a cada elemento del universo de discurso el grado con que pertenece al conjunto difuso. Las funciones más comunes son: Gamma, L, Lambda (triangular) y Pi (trapezoidal).
- *Conjunto difuso*: Valor lingüístico junto con una función de pertenencia.

2.2. Ingeniería del conocimiento

Al proceso de construir un SE se le denomina Ingeniería del Conocimiento [7], y consiste en la adquisición de conocimiento a partir de un especialista humano o de otra fuente y su codificación en el Sistema Experto.

El conocimiento extraído de los expertos y de la literatura, puede evolucionar durante el proceso de desarrollo, en el uso del sistema, por nuevos conocimientos del dominio o por consideraciones de los expertos humanos. Para desarrollo de un SE no es aplicable un ciclo de vida clásico ya que se requiere de un mantenimiento continuo de la base de conocimientos [9]. En la Figura 1, se muestran los tres procesos fundamentales de la Ingeniería del Conocimiento [10].

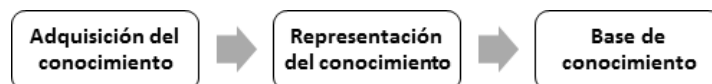


Fig. 1. Ingeniería del conocimiento de manera general.

En la *primera etapa* se tiene acercamiento con los expertos humanos para el dominio del conocimiento. La *segunda etapa* comprende la representación del conocimiento, en la cual el ingeniero del conocimiento codifica y se hacen explícitas las reglas para que el experto humano sea capaz de resolver problemas reales. En la *tercera etapa* la información puede ser representada por medio de reglas de producción (implicaciones lógicas).

2.3. Representación del conocimiento

La base del conocimiento es una colección de datos que representan conocimiento, dicha base se plasma utilizando un lenguaje de representación del conocimiento, y un programa para ampliar y/o consultar una base de conocimiento.

A partir del conocimiento existente puede crearse nuevo conocimiento utilizando la inferencia lógica [11].

$$\begin{aligned} \text{información} &= \text{datos} + \text{significado} \\ \text{conocimiento} &= \text{información} + \text{procesamiento} \end{aligned}$$

Además de su capacidad para hacer abstracciones a diferentes niveles, el conocimiento se relaciona con la acción. El uso efectivo del conocimiento conduce a la formación de planes de acción y finalmente a una profunda comprensión.

El lenguaje estructurado es el más popular para la realización de la representación del conocimiento; incluye una base en lógica de primer orden, reglas de producción, representaciones basadas en frames, combinaciones de frames y la lógica. Las reglas de producción han sido las estructuras más comunes para representar el conocimiento en los SE [12].

3. Diseño de la base de conocimiento

El diseño de la base de conocimiento para el SED propuesto, consiste en emplear las herramientas y métodos de la Ingeniería del Conocimiento, descritos en los siguientes pasos:

1. **Definición del alcance y adquisición del conocimiento:** Los dos elementos más importantes para la obtención del conocimiento son el experto humano y el ingeniero del conocimiento. Se respetaron las etapas de adquisición del conocimiento presentadas en [10], que pueden observarse en la Figura 2.
 - a) *Identificación:* Se realiza una entrevista no estructurada, en la que, el ingeniero de conocimiento plantea el problema y el experto (médico) presenta las características principales. Se aprende sobre la situación del problema y se plasman por escrito dichas características.
 - b) *Entendimiento:* Con la finalidad de deducir las posibles representaciones de la base de conocimiento, se continúa con la consulta de bibliografía médica: manuales del IMSS, artículos científicos y observaciones a personas con diabetes.
 - c) *Formalización:* Durante esta etapa, se organiza el conocimiento, se representa con una matriz de conocimiento, posteriormente con una red semántica y finalmente con reglas de producción. Estas representaciones se depuran a través de entrevistas estructuradas entre el ingeniero del conocimiento y el experto.
 - d) *Implementación:* Para la realización del prototipo, se emplea el lenguaje Java, se mapean las reglas de producción mediante jFuzzyLogic (librería de código abierto que permite diseñar controladores lógicos difusos)[16] y se generan los métodos del Servicio Web XML.
 - e) *Pruebas:* Se realizan ejecuciones del SED con los síntomas de 20 pacientes.

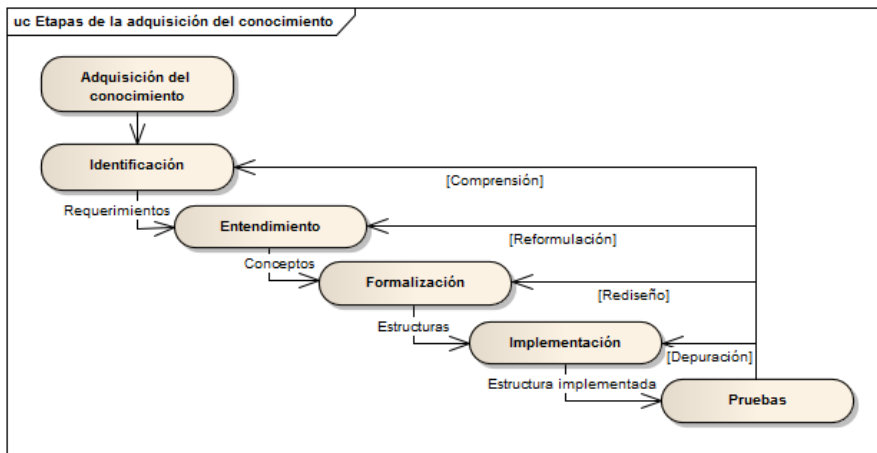


Fig. 2. Etapas de la adquisición del conocimiento.

Durante la fase de identificación al igual que en la fase de entendimiento, se definió el alcance que se tendría con el SE, y la formulación del conocimiento fundamental [13]. El dominio comprende el diagnóstico de cuatro tipos de diabetes: prediabetes, DM I, DM II y diabetes gestacional. La determinación del diagnóstico se realiza de acuerdo a las 21 variables establecidas por el experto humano junto con material bibliográfico; estas variables corresponden a los principales síntomas y características de las personas al momento de realizar un diagnóstico de diabetes.

2. **Matriz de conocimiento:** En la Tabla 1, se definen las 21 variables: binarias (B) y difusas (D), y se identifican cuáles son aplicables a cierto tipo de diabetes.
3. **Red semántica:** Para la representación del conocimiento, se empleó una red semántica, cuyos arcos están dirigidos. Los nodos representan las variables de la Tabla 1.
4. **Reglas de producción:** La codificación de la red semántica, se realizó a través de reglas de producción en jFuzzyLogic, supervisadas por el ingeniero de conocimiento y el experto, que consideran las condiciones y decisiones. Se emplea el método de encadenamiento hacia delante[4], ya que la determinación del tipo y nivel de diabetes se hace a través de un cuestionario. Las 10,000 reglas de producción consideran las variables difusas y las binarias. A continuación se muestra como ejemplo, el diseño de las reglas de producción para la determinación de la presión del paciente, se consideran la presión diastólica (pad) y la presión sistólica (pas).

```

RULE 1 : IF (pad IS normal AND pas IS normal) THEN resultado IS normal;
RULE 2 : IF (pad IS normal AND pas IS medio) OR (pad IS medio AND (pas
IS normal OR pas IS medio)) THEN resultado IS medio;
RULE 3 : IF ((pad IS normal OR pad IS medio OR pad IS alto) AND pas IS
alto) OR (pad IS alto AND (pas IS normal OR pas IS medio)) THEN
resultado IS alto;
    
```

Tabla 1. Variables definidas para cada tipo de diabetes.

No.	Criterio	Tipo	Prediabetes	DMI	DM2	Gestacional
1	Sexo	B	X	X	X	X
2	IMC	D	X		X	
3	Edad	D	X	X	X	X
4	Embarazo	B				X
5	Glucosa en ayunas	D	X	X	X	X
6	Tolerancia a glucosa	D	X	X	X	X
7	Antecedentes familiares	B		X	X	X
8	Sedentaria	B		X	X	
9	Consume alimentos grasosos	B		X	X	
10	Cansancio	B		X	X	X
11	Pérdida de peso	B		X		
12	Aumento apetito	B		X	X	X
13	Colesterol	D		X	X	
14	Presión sistólica	D		X	X	X
15	Presión diastólica	D		X	X	X
16	Hidratación	D		X	X	X
17	Micción	D		X	X	X
18	D. gestacional previa	B				X
19	Parto previo con sobrepeso	B				X
20	Bebé previo con malformaciones	B				X
21	Vómito	B				X

4. Implementación

Esta sección describe el desarrollo del SE difuso para el diagnóstico de diabetes. En la Figura 4, se presenta la arquitectura del SED propuesto. A continuación se describen los elementos de esta arquitectura:

- *Aplicación Web*: Es el mecanismo mediante el cual, el médico introducirá los datos y síntomas de sus pacientes, esta información la podrá consultar posteriormente con su usuario y contraseña. La aplicación enviará su solicitud al servicio web, que se comunicará con el SED y se realizará el fuzzificado de las entradas en el motor de inferencia para posteriormente retornar una respuesta de diagnóstico. La aplicación está desarrollada en PHP. La interfaz de la aplicación para el diagnóstico de DMI se muestra en la Figura 3; para cada tipo de diabetes se muestra una interfaz correspondiente con sus síntomas.
- *Servidor Web*: Se usó el Servidor GlassFish para el funcionamiento de los Servicios Web implementados con JAX-WS.
- *Base de datos*: Almacenará la información de entrada y salida del Servicio Web (síntomas y diagnósticos).
- *Servicio Web*: Emplea el estándar XML para realizar un intercambio de datos con los clientes desarrollados en diferentes plataformas: Aplicación Web y/o móvil. El servicio web cuenta con siete métodos (login, datosPersonales,

glucosaAyunas, toleranciaGlucosa, DMI, DMII y DGestacional) que reciben los datos y síntomas para dar una respuesta según el análisis del motor de inferencia.

- *Sistema Experto Difuso*: Sistema que cuenta con el conocimiento definido en la Sección 3.

Sistema Experto para Diagnóstico de Diabetes

1 Ingrese sus datos personales

Nombre Femenino

Estatura (en metros) Peso (en kilos)

Edad No embarazada

*Todos los campos son obligatorios

Continuar

2 Ingrese el valor de glucosa en ayunas

Glucosa en ayunas

Continuar

3 Ingrese el valor de tolerancia a la glucosa

Tolerancia oral a la glucosa

Continuar

4 Diagnóstico de Diabetes Tipo 1

No antecedentes familiares Persona no sedentaria

Sin consumo alto de grasas y carbohi Sin cansancio durante el día

Sin pérdida de peso sin razón Sin aumento de apetito

Nivel de colesterol Presión sistólica

Presión diastólica Litros de agua al día

Orina durante el día (litros)

Continuar

5 Resultado del diagnóstico

Diabetes Mellitus tipo 1 en un 89%

Recomendaciones: Repartir sus calorías en: Carbohidratos 50%, proteínas 15 - 20% y grasas (aceites vegetales) 30 - 35%. Incorporar fibra en su alimentación. Consumir fruta fresca. Realizar al menos 30 minutos de ejercicio, como trotar. Hidratarse al menos con 2.5 litros de agua.

Fig. 3. Interfaz de la Aplicación Web, (1) Datos generales del usuario, (2) Valor de glucosa en ayunas, (3) Valor de tolerancia a la glucosa, (4) Datos para diagnóstico de DMI, (5) Resultado del diagnóstico y recomendaciones.

4.1. Sistema experto difuso

En esta sección se describen los elementos para modelar el SED.

1. *Fuzzificación*: Proceso para convertir las variables en valores difusos, para ello, se utilizó la función de pertenencia triangular (1). Los parámetros de las funciones se muestran en la Tabla 2.
2. *Proceso de inferencia*: Las operaciones lógicas aplicables a los conjuntos difusos, estas son utilizadas por el motor de inferencias para derivar un resultado, tal y como lo hace el cerebro humano, utilizando el razonamiento. El motor de inferencias es implementado usando jFuzzyLogic; las variables de entrada son definidas en reglas y almacenadas en un archivo FCL que será procesado por la librería.

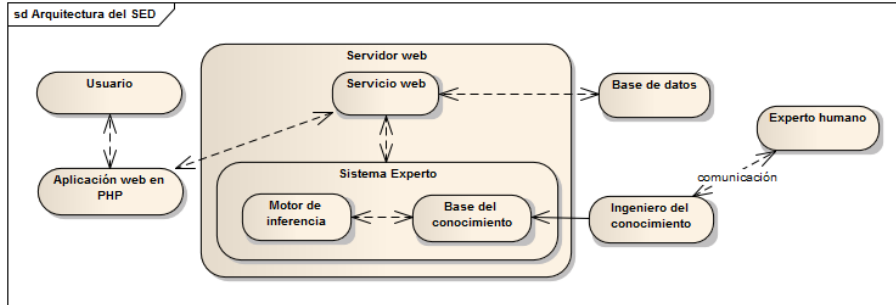


Fig. 4. Arquitectura del sistema experto difuso.

3. *Defuzzificación*: Es el proceso contrario a la fuzzificación; convierte el valor difuso en un valor nítido. Los métodos más habituales de defuzzificación son: centro de gravedad, el centro de sumas y la media de los máximos [3]. El SED utiliza método del centro de gravedad (2).

$$\text{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \quad (1)$$

$$z^{\text{COA}} = \frac{\int_z \mu_A(z)z dz}{\int_z \mu_A(z) dz} \quad (2)$$

5. Experimentos y resultados

El sistema fue probado en un grupo de veinte pacientes de la Unidad de Medicina Familiar del ISSSTEP: 3 hombres con DMI, 3 mujeres con DMI, 5 hombres con DMII, 3 mujeres con DMII, 2 mujeres con PD, 3 mujeres adolescente con DG y 1 mujer adulta con DG. La Figura 5 a) muestra los tiempos que les tomó a los pacientes en su consulta tradicional y con el SED. La Figura 5 b) muestra el número de síntomas solicitados por el SED y los de una consulta tradicional.

Con los resultados mostrados en la Figura 5, se puede observar que el SED ayuda considerablemente a los médicos a determinar un diagnóstico en un menor tiempo al que les toma comúnmente y que además, se consideran más síntomas por parte del SED promoviendo un mejora en la productividad de los médicos. Los médicos lograron complementar las recomendaciones proporcionadas por el SED y aconsejaron a sus pacientes en tratamiento farmacológico para un mejor control de glucosa.

Tabla 2. Parámetros de funciones de pertenencia triangular.

Num	Variable lingüística	Valor lingüístico	Universo de discurso
1	IMC	Deseable	[18.5, 25, 30]
		Intermedio	[25, 30, 35]
		Elevado	[30, 35, 45]
2	Edad	Niño	[5, 9, 12]
		Adolescente	[10, 15, 20]
		Adulto	[20, 30, 50]
		Adulto mayor	[44, 60, 100]
3	Glucosa en ayunas	Normal	[, 50, 79]
		Óptimo	[60, 98, 126]
		Prediabetes	[100, 125, 130]
		Diabetes	[112, 146, 162]
4	Tolerancia a glucosa	Normal	[56, 138, 145]
		Intolerancia	[138, 170, 202]
		Alto	[195, 202, 250]
5	Colesterol	Deseable	[, 50, 200]
		Intermedio	[180, 216, 245]
		Elevado	[235, 280, 300]
6	Presión sistólica	Normal	[, 90, 130]
		Medio	[120, 130, 160]
		Alto	[130, 250,]
7	Presión diastólica	Normal	[, 60, 85]
		Medio	[80, 85, 100]
		Alto	[85, 140,]
8	Hidratación	NormalNiña	[1.3, 1.6, 2]
		AltoNiña	[1.9, 2.5, 3]
		NormalNiño	[1.3, 1.8, 2.2]
		AltoNiño	[2.1, 2.7, 3.2]
		NormalAdolescenteMujer	[1.5, 1.9, 2.3]
		AltoAdolescenteMujer	[2.1, 2.7, 3.2]
		NormalAdolescenteHombre	[2.1, 2.6, 3]
		AltoAdolescenteHombre	[2.8, 3.4, 3.8]
		NormalAdultoMujer	[2, 2.2, 2.6]
		AltoAdultoMujer	[2.5, 3, 3.4]
		NormalAdultoHombre	[2.5, 3, 3.4]
		AltoAdultoHombre	[3.2, 3.7, 4.1]
9	Micción	NormalNiña	[0.6, 0.7, 1]
		AltoNiña	[0.9, 1.75, 2.1]
		NormalNiño	[0.6, 0.8, 1.1]
		AltoNiño	[1.0, 1.8, 2.2]
		NormalAdolescenteMujer	[0.7, 1.1, 1.5]
		AltoAdolescenteMujer	[1.4, 2.5, 3.0]
		NormalAdolescenteHombre	[0.8, 1.5, 1.9]
		AltoAdolescenteHombre	[1.4, 3.5, 4.5]
		NormalAdultoMujer	[1.1, 1.5, 2]
		AltoAdultoMujer	[1.9, 3, 4.5]
		NormalAdultoHombre	[1.1, 1.7, 2.1]
		AltoAdultoHombre	[2, 4, 5]

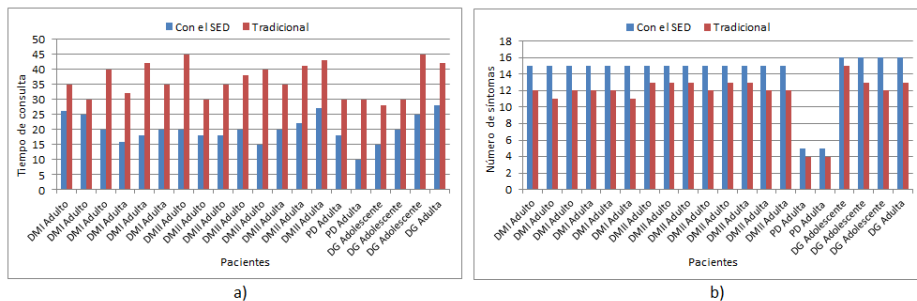


Fig. 5. a) Tiempos de los pacientes en su consulta tradicional y con ayuda del SED. b) Síntomas solicitados en consulta tradicional y con ayuda del SED.

6. Conclusiones y trabajo futuro

Este trabajo muestra un apoyo en el diagnóstico de diabetes. El sistema está descrito como una integración de un SED, Servicio Web y Aplicación Web. Partiendo de que la base del conocimiento debe ser lo más completa posible, el diseño de la base de conocimiento fue desarrollado siguiendo la metodología de la Ingeniería del Conocimiento. En comparación de algunos sistemas expertos propuestos para el diagnóstico y detección de diabetes [14, 15], se presenta una base de conocimiento más basta, en la que se consideran más variables de cada tipo de diabetes para un mejor diagnóstico. Así también, se tomó en cuenta la presencia de diabetes gestacional en cualquier edad de la paciente.

Con la información de las pruebas realizadas al SED, el médico podrá tener un mejor historial de las consultas al alcance de su mano, ya que la información estará disponible las 24 horas del día, a través de la Aplicación Web diseñada, además, le permitirá tener una retroalimentación en las siguientes consultas a sus pacientes.

Como trabajo futuro se ampliará el conocimiento con un nutriólogo, con el propósito de dar recomendaciones no farmacológicas específicas de alimentación y ejercicio a los pacientes. Además se desarrollarán aplicaciones cliente en dispositivos móviles (iOS y Android), extendiendo así la posibilidad de que los médicos sin tanta experiencia puedan ofrecer un diagnóstico más certero en cualquier área geográfica que cuente con acceso a internet y que pueda enviar el resultado del diagnóstico al e-mail del paciente.

Referencias

1. Coiera, E.: Guide to Medical Informatics, the Internet and Telemedicine. London, New York, Chapman & Hall Medical (1997)
2. Instituto del Seguro Social: Guía de Práctica Clínica GPC, Diagnóstico y Tratamiento de la Diabetes Mellitus tipo 2. México (2012)

3. Márquez-Hernández, F.A.: Cooperación entre sistemas de inferencia, métodos de defuzzificación y aprendizaje de sistemas difusos lingüísticos. Tesis doctorado, Benemérita Universidad Autónoma de Puebla, México (2013)
4. Giarratano, J., Riley, G.: Sistemas expertos: principios y programación. International Thomson, México (2001)
5. Quiroz-Hernández, J.L.: Prototipo de un sistema experto en el diagnóstico de acné. Tesis licenciatura, Benemérita Universidad Autónoma de Puebla, México (2000)
6. Cardona, C., Restrepo, C., Padilla, S.: Diseño y desarrollo de un sistema experto con lógica difusa para diagnosticar el nivel psicológico, social y criminalístico de individuos de acuerdo a su prospección a cometer ciertos delitos. Proyecto Curso Inteligencia Artificial, Universidad Nacional de Colombia, Medellín (2006)
7. Michie, D: Knowledge Engineering. In: Kybernetes. Gordon and Breach Science Publishers Ltd, Vol. 2, pp. 197–200, Inglaterra (1973)
8. Conde-Ramírez, J.C.: Modelado cognitivo en videojuegos. Tesis maestría, Benemérita Universidad Autónoma de Puebla, México (2013)
9. Fernández-Fernández, G.: Representación del conocimiento en sistemas inteligentes, disponible en: <http://www.gsi.dit.upm.es/~gfer/ssii/rcsi/>. Universidad Politécnica de Madrid, España (2004)
10. Haces-Álvarez, J.A.: Sistema para generar micromundos para la asignatura de álgebra. Tesis licenciatura, Universidad Nacional Autónoma de México, México (2011)
11. Free On-Line Dictionary of Computing (FOLDOC): Knowledge. Disponible en: <http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=knowledge>
12. Biondo, S.: Fundamental of expert systems technology Principles and concepts. Ablex Publishing Corporation, EUA (1990)
13. Aristizábal-Mejía, N., Torres-Moreno, M.E.: Técnicas de levantamiento de requerimientos con innovación. En: Cuarto Congreso Colombiano de Computación 4CCC, Sociedad Colombiana de Computación S(Co)2. Colombia (2009)
14. Osorio-Vega, C.F.: Sistema experto para determinar el tipo de diabetes. Tesis licenciatura, Universidad Tecnológica Equinoccial, Ecuador (2009)
15. Rey-Salazar, G., García-Araya, A.: Sistema experto para determinar tipo de diabetes. Ingeniería Civil Electrónica, Departamento de Ingeniería Eléctrica, Universidad de La Frontera. Chile (2007)
16. Cingolani, P., Alcalá-Fdez, J.: jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming. In: International Journal of Computational Intelligence Systems, Vol. 6, Supplement 1, pp. 61–75 (2013)

Using the M-Learning to Support Learning of the Concept of the Derivative

Daniel A. Sarmiento, Chadwick Carreto A, Elena F. Ruiz

Instituto Politécnico Nacional, School of Computing (ESCOM),
Mexico

{dasarmiento, ccarreto, eruiz}@ipn.mx

Abstract. One of the main obstacles in Mexico's engineering programs is math comprehension, especially in the Derivative concept. Due to this, we present a study case that relates Mobile Computing and Classroom Learning in the "Escuela Superior de Cómputo", based on the Educational model of the Instituto Politécnico Nacional (competence based work and problem solutions) in which we propose apps and activities to teach the concept of the Derivative. M-Learning is emphasized as one of its lines, as the objective is the use of mobile devices running an app that uses its components such as sensors, screen, camera and processing power in classroom work. In this paper, we employed Augmented Reality (ARRoC), based on the good results this technology has had in the field of learning. This proposal was developed using a qualitative research methodology supported by quantitative research. The methodological instruments used on this proposal are: observation, questionnaires, interviews and evaluations. We obtained positive results with a 40% increase using M-Learning, from the 20% increase using traditional means.

Keywords: Augmented reality, classroom learning, educational research, mobile computing.

1 Introduction

At the time of writing, there are software programs that have the purpose to help math comprehension. Nevertheless, examining this programs yields that they only develop algebraic and memory competences, leaving aside visualization, comprehension and reasoning in math [1].

One of the major shortcomings of this programs is that they cannot be executed in mobile devices and those that can are have limited features, lacking a teacher's classroom support. Due to the low number of mobile applications that develop different competences and that combine classroom learning and M-Learning [2], we propose the use of mobile computing through an educational application, having as a study case the topic of Derivative corresponding to ESCOM's curricula.

A. Mobile Computing

Mobile computing is being developed, explored and use in different countries, because mobile applications allow the student to build their own learning environment with all the necessary resources and tools in a single device. Technology has evolved from simple phones to handheld computers that are capable of browsing the internet, checking email, using location services (GPS) and that have complex sensors such as a multitouch screen, digital compass and accelerometer. Mobile software has also evolved to a point that resembles desktop operating systems.

B. Instituto Politécnico Nacional's Educational Model

The Polytechnic Institute's Educational Model promotes an integral, high quality formation, that's oriented towards the student's learning. To achieve this, it is necessary to have flexible programs that incorporate the possibility of transit between modalities, programs, levels, academic units and the diversification of learning spaces. It is also necessary to introduce teaching methodologies that prioritize innovation, creative capacity and the use of Information and Communication Technologies.

C. Mobile Learning

Mobile devices applied to educational environments are an innovative alternative that can potentially help improving teaching-learning processes [10]. We denominate the intersection between online education and mobile computational devices as Mobile Learning [2]. It offers advantages such as frequent and integral access to applications that support learning anytime anywhere. In other words, mobile learning can be seen as the use of mobile devices in the learning process.

D. Augmented Reality

Augmented Reality is the term used to define a direct or indirect vision of a physical environment of the real world, in which its elements are combined with virtual elements to create a mixed reality in real time [11, 12].

Augmented reality is made of a real scene, which is obtained in real time using a device's camera and the additional information associated to that scene. The fusion of both elements is made using a mobile device's computing power to show a single image on the screen. The use of such technologies in a school must be framed by its educational model, so that they can be easily used in classrooms.

2 Objectives

As we mentioned before, the main problem lies in using applications that develop a range of competences in the student with a model that promotes the use of Information and Communication Technologies. Thus, we have the following main objective:

Evaluate the combination of mobile computing and classroom learning on the Derivative case study as an education support tool. From this main objective we derive the following secondary objectives:

- Determine the relevance of using Mobile Computing and classroom learning.

- Establish Mobile Computing activities and apps that support learning the Derivative concept.
- Verify the efficiency of mobile computing combined with classroom learning.

3 Methodological Aspects

Our object of study are students of the first semester in the “Escuela Superior de Cómputo”, given that the Calculus Learning Unit, which contains the Derivative as one of its topics, is taught on that semester. It is important to emphasize that the qualitative approach helps document and analyze the processes that use Mobile Computing [1]. This approach must be based on a qualitative research in which we assume that the meaning of concepts and processes development are socially built in practice [13]. It is also necessary to select a sample, which allows us to conduct the study in an efficient way. Thus, the research methodology used is of a mixed character, because we employed a qualitative methodology that gathers data without numerical measurement and a qualitative to represent the results. The instruments we used are:

- Observation.
- Questionnaires.
- Interviews.
- Evaluation of the obtained results.

The instruments above are used throughout the research and will be described on the “Case Study” section. The proposed stages of proposed research methodologies, instruments and teaching methodology are executed in parallel. We then propose the following stages for the research methodology:

1. Perform diagnostic evaluations and surveys to the study group.
2. Analyze the Learning Unit’s activities.
3. Develop classroom and mobile learning activities.
4. Gather and analyze the activities’ results.

The stages for the methodological instruments are as follow:

1. Select data instruments such as observation, questionnaires, surveys, interviews and study subject.
2. Analyze the functionality of the data gathering instruments to select the software methodology.
3. Carry out the Software Engineering process.
4. Analyze the results obtained by the Software Engineering methodology.

The teaching methodology is divided in three: materials, resources and results and subdivided in the following stages.

Materials and resources stages:

1. Select materials such as pencils, paper sheets, notebooks, blackboard and mobile device.
2. Use mobile devices on the study group.
3. Analyze the resources needed to deploy a server or platform if necessary.

4. Analyze the total budget spent in resources and materials.

Results stages:

1. Results given by school statistics and activities' requirements.
2. Results given by activities in classroom and using mobile devices.
3. Recollecting the total results given by the previous phases.
4. Answering the research questions and validating the research hypothesis.

A. Survey on Mobile Devices

We employed a predesigned survey as a methodological instrument and observational study that allows the researcher to gather data without modifying the environment nor controlling the observed process. The data was obtained using a set of normalized questions directed towards a representative sample of the case study, which is composed by ESCOM's students [1].

A closed survey was selected due to it being of uniform character and easier to quantify. In this kind of survey both, study group A and study group B select just one of the options given from a list of possibilities.

The questions and answers were previously formulated by the researcher, thus allowing to obtain the needed information.

B. Diagnostic questionnaire

A diagnostic questionnaire as a methodological instrument is useful in gathering data and as a starting point for the use of qualitative methods [14, 15]

A questionnaire must be clear, unambiguous and uniformly accomplishable. In this case, the diagnostic questionnaire was made of ten questions related to the Calculus Learning Unit, specifically to the Derivative concepts. These questionnaires were previously validated [16, 17]. We selected two study groups, denominated A and B, in which former had high academic performance and the latter had an average performance.

Each question was classified as algorithmic or concept related.

- Algorithmic questions are those in which the student must know the method or the steps needed to solve the questions and are only asked to calculate or find a result.
- Concept related questions imply that the student understands the concept involved so that he/she can find the correct answer.

The activities that were performed in group A and B were designed based on the questions of the diagnostic questionnaires. For this purpose, we selected the questions with a higher rate of failure, which were the ones that asked them to calculate the slope of a tangent line and the rate of change.

C. Activity 1. Derivate concept

Figure 2 shows the definition of the derivative as a tangent line to a curve [18], then it describes its purpose.

D. Application 1. SENSORTAG

The SensorTag activity was designed to get the inclination angle of a slope using the mobile device's sensors. The survey we performed showed that 97.9% of the mobile devices use by both study group A and B had the capability to run this activity. Figure 3 shows one of the views of the SensorTag App.

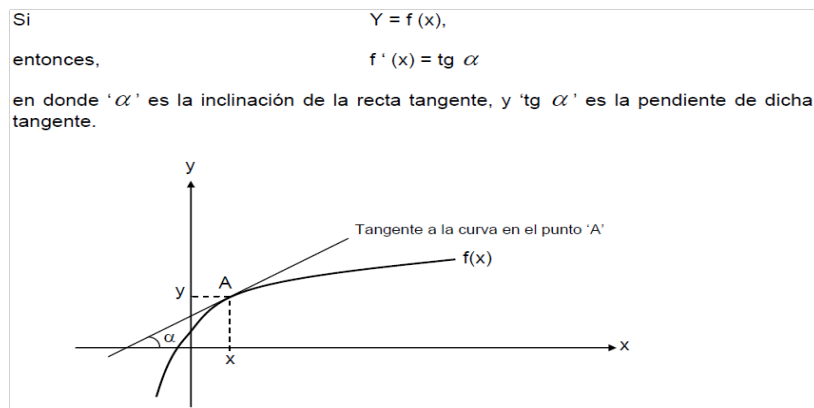


Fig. 2. Geometrical definition of a derivative.



Fig. 3. SensorTag activity.

E. Activity 2. Rate of Change

Figure 4 shows an example in which the rate of change concept is applied.

Purpose: We wish to determine if the student comprehends the concept of the rate of change, using an example in which an Android shaped balloon loses gas, radius and surface as time goes by.

F. Application 2. ARRoC

The ARRoC activity was designed so that group A and B could see in the classroom what happens in real time to an object that suffers a variation in an instant, which is known as rate of change and can be obtained using the derivative.

El gas de un globo esférico en forma de Android se escapa a razón de $1,000 \text{ cm}^3/\text{min}$ en el mismo instante en que el radio es de 25 cm .
a) ¿Con qué rapidez disminuye el radio?
b) ¿Con qué rapidez disminuye el área de la superficie?

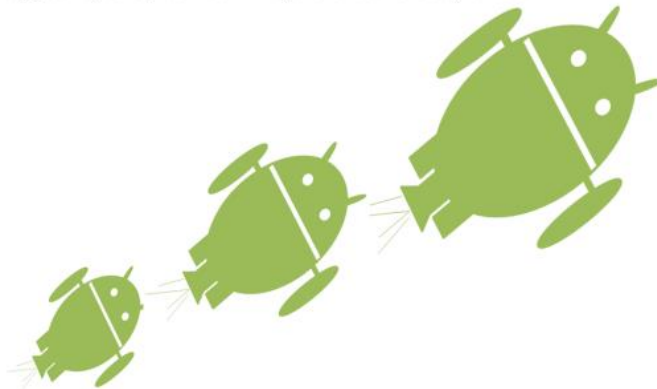


Fig. 4. Rate of change example.

This activity uses the mobile device's camera and an augmented reality marker, which allows to visualize the object. Figure 5 shows a screenshot of the running app.

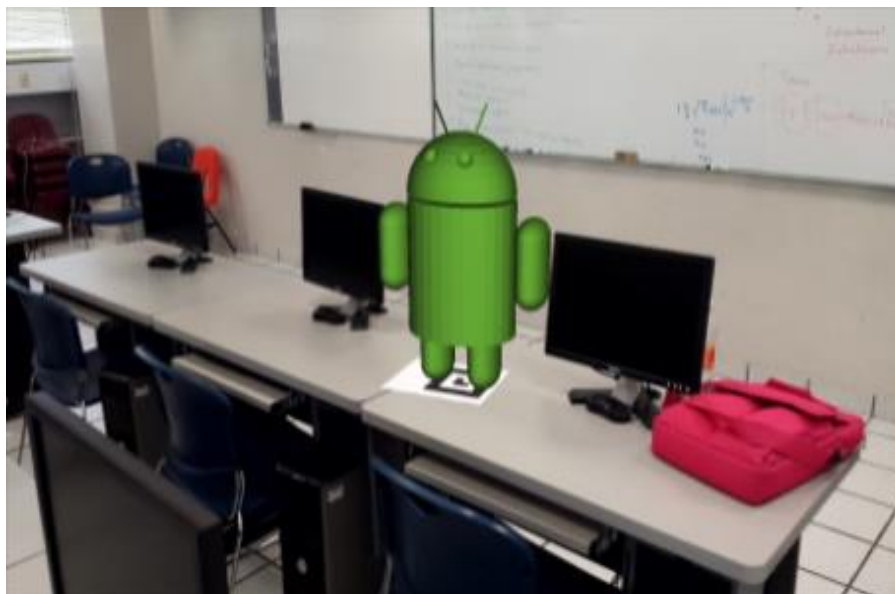


Fig. 5. ARRoC visualization.

4 Results

The surveys conducted in ESCOM show that 80% of the students possess a smartphone, which has a great computing power. The specifications of this mobile devices allow us to take advantage of the similarities between how the students use their mobile devices and the National Polytechnic Institute's Educational Model.

It is also worth mentioning that when the teacher made the application available to the students, they showed interest in using their smartphone and had better grasp of the derivative concept.

The results obtained by study group A, showed that the students had an acceptable performance, responding well in an overall way. The same phenomenon was observe in study group B, which had an excellent performance in the evaluation questionnaire as they all passed it. This results are of great importance, because they show the difference between the diagnostic questionnaire in which just a minimum amount of students had an approbatory grade.

Tables 1 and 2 show each question's results divided by each study group.

Table 1. Results of study group A.

SensorTag Application Questions	Questions	Correct percentage
How should the derivative of a function be when the tangent line has a positive slope?	1	87%
How should the derivative of a function be when the tangent line has a negative slope?	2	87%
How should the derivative of a function be when the tangent line has a value of 0?	3	83%
What can we obtain when the derivative of the slope has a value of 0?	4	100%
ARRoC Application Questions		
How fast does the radius of the android's arms decrease?	1	100%
How fast does the area decrease on the surface of the android's arms?	2	83%

Table 2. Results of study group B.

SensorTag Application Questions	Questions	Correct percentage
How should the derivative of a function be when the tangent line has a positive slope?	1	93%
How should the derivative of a function be when the tangent line has a negative slope?	2	94%
How should the derivative of a function be when the tangent line has a value of 0?	3	88%
What can we obtain when the derivative of the slope has a value of 0?	4	100%

ARRoC Application Questions	Questions	Correct percentage
How fast does the radius of the android's arms decrease?	1	100%
How fast does the area decrease on the surface of the android's arms?	2	86%

5 Discussion

Mobile computing is the best tool for the student to visualize situations where the derivative concept is present. Even though a graphing calculator could be use, it's more feasible to use a mobile device, since most students already possess one and a mobile device has more features like internet capabilities, sensors and a camera; which help out the student during his/her lessons.

6 Conclusion

After examining the developed applications and activities in this research, we observed that Mobile Computing is present at all times in all places.

We determined the specifications of the mobile devices owned by the case study students and we conclude that Mobile computing and classroom learning is relevant in the Derivative learning process.

Working with a qualitative methodology supported by quantitative aspects helped us have a better vision of what happened during work sessions, prioritizing observation, which helped us validate the development of this research. The Mobile Computing's activities, were designed and implemented. This activities helped the students learn the Derivative concept by allowing the development of competences such as the ability to visualize phenomena, reflection capacity and problem solving abilities.

This allowed us to find a relationship between mobile Computing and Educational Mathematics, emphasizing aspects that are outlined in the National Polytechnic Institute's educational model. Using mobile computing, students could transition from concrete concepts to abstract concepts, generating tangible products, which allowed them to have a clear idea of the concepts given to them in the case study.

After analyzing the combination of Mobile Computing and Classroom Learning we conclude that Mobile computing is a valuable tool in supporting the teaching process, as showed by the 40% increase in the topic understanding, opposed to the 20% average increase by using traditional means. This result proves that the research hypothesis was correct. Using mobile computing in the classroom allows decreasing mechanical solutions to problems, visualizing and proposing solutions, and motivates students to use their mobile devices as information capturing devices; helping them process information and thus, reaching valid conclusions.

Finally, we observed that mobile devices are replacing calculators and computers in classrooms, with the clear advantage of portability and real time information capturing.

Acknowledgment. We want to thank Secretaría de Investigación y Posgrado (SIP) of the Instituto Politécnico Nacional for the support during this work, project 20150296.

References

- 1 Hernández Sampieri, R., Fernández Collado, C., P. Baptista, L.: Metodología de la Investigación. McGraw-Hill, México, pp. 350–500 (2008)
- 2 Potts, J., Moore, N., Sukittanon, S.: Developing Mobile Learning Applications for Electrical Engineering Courses. University of Tennessee at Martin, Department of Engineering, TN. U.S.A., pp. 2–6 (2011)
- 3 Secretaría Académica, Instituto Politécnico Nacional IPN, Dirección de Educación Superior, Escuela Superior de Cómputo (ESCOM): Programa Sintético, Plan 2009 Unidad de Aprendizaje de Cálculo. <http://148.204.58.221/DISC/temarios/plan2009/NIVEL2/Calculo.pdf>. 21 de Noviembre de 2013 (2009)
- 4 Secretaria de Educación Pública. Instituto Politécnico Nacional. Mapa Curricular ESCOM, <http://www.escom.ipn.mx/Oferta/Paginas/mapacurricular.aspx> (2013)
- 5 MacOS. Apple Developers. Desarrolladores de Apps, <https://developer.apple.com/> (2013)
- 6 Nestel, D., Ng, A., Grey, K., Hill, R., Villanueva, E., Kotsanas, G., Oaten, A., Browne, C.: Evaluation of Mobile Learning: Students Experiences in a New Rural-Based Medical School, BMC Medical Education, <http://www.biomedcentral.com/1472-6920/10/57> (2010)
- 7 Google. Android Developers. Desarrolladores de Aplicaciones. <http://developer.android.com/index.html> (2013)
- 8 MacOS, Apple Developers. Herramienta de Desarrollo IDE. <https://developer.apple.com/xcode/> (2013)
- 9 Instituto Politécnico Nacional. Materiales para la Reforma: Un Nuevo Modelo Educativo para el IPN. Talleres Gráficos de la Dirección de Publicaciones del Instituto Politécnico Nacional, D.F., México, P18 (2004)
- 10 Johnson, L., Adams, B., Cummins, S., Estrada, M., Freeman, V., Ludgate, A.: Higher Education Edition 2013. Austin. Texas: The New Media Consortium, pp. 2–15 (2013)
- 11 Raghav, S.: Pro Android Augmented Reality, New York, Apress, pp. 1–150 (2012)
- 12 Ariel y Fundación Telefónica con colaboración del Banco Interamericano de Desarrollo. Realidad Aumentada un Nuevo Lente para Ver el Mundo. Ariel ed. Madrid, España, pp. 22–80 (2011)
- 13 Moschkovich, J., Brenner, M.: Integrating a Naturalistic Paradigm into Research on Mathematics and Science Cognition and Learning. In: Handbook of Research Design in Mathematics and Science Education, Erlbaum Associates Inc., pp. 457–486 (2000)
- 14 Valdemoros Álvarez, M.E.: La Constancia de la Unidad en la Suma de Fracciones: Estudio de Caso. Investigaciones en Matemática Educativa II, pp. 465–481 (1998)
- 15 Valdemoros Álvarez, M.E.: Recursos Intuitivos que Favorecen la Adición de Fracciones: Estudio de Caso. Educación Matemática, pp. 5–17 (1997)

- 16 Gutiérrez G., J.J.: Sistema Móvil como Herramienta de Apoyo para el Aprendizaje de Cálculo. Caso de Estudio Funciones. Tesis, Escuela Superior de Cómputo (ESCOM), IPN, México DF, México, pp. 8–110 (2013)
- 17 Viveros, K.: Continuidad y la Derivada de una Función: Concepciones y Conexiones Intuitivas en Estudiantes Universitarios. Tesis, CINVESTAV-IPN, México DF, pp. 65–90 (2001)
- 18 Jame, S.: Aplicación de la Derivada en Cálculo. Thomson Learning, México, pp. 276–279 (2002)

3-approximation Algorithm for the Travelling Repairman Problem with Unit Time-windows

Luis Eduardo Urbán Rivero¹, Cynthia A. Rodríguez Villalobos²,
Rafael López Bracho³, Francisco Javier Zaragoza Martínez³

¹ UAM Azcapotzalco, Posgrado en Optimización,
Mexico

² University of Waterloo, Department of Combinatorics and Optimization,
Canada

³ UAM Azcapotzalco, Departamento de Sistemas,
Mexico

lurbanrivero@gmail.com, ca7rodri@uwaterloo.ca, rlb@correo.azc.uam.mx,
franz@correo.azc.uam.mx

Abstract. The travelling repairman problem (TRP) is a scheduling problem in which a repairman must visit locations to perform some task. Each location has a time window in which the repairman is allowed to arrive. The objective of this problem is to maximize the number of tasks performed. In this paper, we consider a special case in which all the locations are on a straight line, the tasks have no processing time, and all time-windows are of unit length. We present an improvement on the analysis of the 4-approximation algorithm presented by S. L. Pérez Pérez *et al.* in 2014.

Keywords: TRP, approximation algorithm, unit time-windows.

1 Introduction

In the travelling repairman problem (TRP), we are given a set of locations the repairman can visit and the time needed to travel between any pair of locations. Each location has a set of tasks to be done by the repairman; each with a fixed processing time and a time-window during which the repairman is allowed to arrive at the location to perform that task. The objective of the problem is to find the route which maximizes the number of tasks completed by the repairman.

In 1992, J. Tsitsiklis [5] proved that deciding whether the repairman can complete k tasks within their time-windows is NP-complete, even if the tasks have no processing time.

In 2005, R. Bar-Yehuda, *et al.* [1] considered the special case in which all location are on a line, there are no processing times, and all time-windows are of unit length. They proposed an 8-approximation algorithm with running time

$O(n^2)$. This algorithm has the disadvantage of double-counting some of the tasks performed. Therefore, they also propose a $(4+\epsilon)$ -approximation algorithm which avoids double-counting but has computational cost $O(n^{8/\epsilon})$. It is worth noting that the computational complexity of this special case is unknown.

In 2012, G. Frederickson and B. Wittman [2] showed that the TRP with unit time-windows on a tree is NP-hard. Also, in [2], they propose a 3-approximation algorithm for that case with running time $O(n^4)$. This result has a better approximation factor than the algorithm proposed by R. Bar-Yehuda *et al.* but has a high computational cost.

In 2014, S. L. Pérez Pérez *et al.* present a 4-approximation algorithm for the TRP with unit time-windows on a line. The algorithm is based on the algorithm of R. Bar-Yehuda *et al.* [1], and improves the computational cost to $O(n^2)$. The algorithm is also faster and simpler than the algorithm proposed by G. Frederickson and B. Wittman [2].

In this paper, we give a different analysis to the algorithm proposed by S. L. Pérez Pérez *et al.* [3] and show its approximation factor is 3. In Section 2 we formally describe the problem. In Section 3 we present the $O(n^2)$ algorithm by R. Bar-Yehuda *et al.* [1] for the TRP with unit time-windows on a line. In Section 4 we present the algorithm by S. L. Pérez Pérez *et al.* [3] and give our analysis for the algorithm improving its approximation factor.

2 Preliminaries

We define the travelling repairman problem with time windows (TRP-TW) as follows.

Definition 1. Let (X, d) be a metric space where X is a set of points and $d : X \times X \rightarrow \mathbb{R}_+$ is a metric. We are given:

- a starting point $x_0 \in X$.
- a set of locations $V = \{x_1, x_2, \dots, x_n\}$, where $x_i \in X$ for all $i \in \{1, \dots, n\}$,
- a set of non-negative processing times $\{t_0 = 0, t_1, t_2, \dots, t_n\}$,
- a set of profits $\{p_0 = 0, p_1, p_2, \dots, p_n\}$,
- a set of time windows $\{\ell_i, u_i : 0 \leq \ell_i \leq u_i, i \in \{1, 2, \dots, n\}\}$

The objective is to find a route $x_0 r_1 r_2 \dots r_k$ maximizing $\sum_{i=1}^k p_i r_i$ such that $r_i \in V$ for all $i \in \{1, 2, \dots, k\}$ and $\ell_i \leq \sum_{i=1}^j d(r_{i-1}, r_i) + t_{i-1} \leq u_i$ for all $j \in \{1, 2, \dots, k\}$.

The set of locations V represent tasks to be performed by the repairman. Note that a point in (X, d) can have multiple locations associated if there are multiple tasks to be performed there. The time-windows give the time span during which the repairman is allowed to arrive to a location for a task.

We consider a special case (Line-TRP-UTW) where X is a line, all processing times are 0, all profits are 1, and all time windows are unitary ($u_i - \ell_i = 1$ for all $i \in \{1, 2, \dots, n\}$).

3 8-approximation Algorithm for Line-TRP-UTW

We describe the algorithm proposed by R. Bar-Yehuda *et al.* [1]. First, the position of each location in V and the time window of the associated task on the position vs time plane. In this representation, each task becomes a vertical line segment of unit length. Note that a curve C on the plane will be a feasible route if and only if it is monotone with respect to time and the tangents to any point on C have angle between 45 and 135 degrees.

Then, the plane is rotated clockwise 45 degrees. Because of this, the representations of the tasks become line segments of length 1 with slope of 45 degrees. and a curve C on the plane is a feasible route if and only if it is monotone with respect to both axes. The Line-TRP-UTW is then equivalent to finding a monotone curve on this plane maximizing the number of intersected line segments.

Afterwards, we overlay a (finite) grid with square size $1/\sqrt{2}$ on the plane such that it does not intersect the extreme point of any of the slanted line segments and all slanted segments lie inside the grid. Note that every slanted segment will intersect the grid at exactly one horizontal line and one vertical line. We then transform the grid to a directed acyclic graph (DAG) $\bar{G} = (V, A)$ by setting all the intersections of grid lines as vertices and orienting all the horizontal edges rightwards and all the vertical edges upwards. Finally, we associate a weight $w(a)$ to every arc $a \in A$ corresponding to the number of line segments intersecting that arc.

The size of \bar{G} is polynomial in $\max\{u_i : i \in \{1, \dots, n\}\} - \min\{u_i : i \in \{1, \dots, n\}\}$ and $\max\{x_i, i \in \{0, 1, \dots, n\}\} - \min\{x_i, i \in \{0, 1, \dots, n\}\}$ but it is not strongly polynomial in the size of the input. However, it is possible to reduce the grid to a strongly polynomial size.

Theorem 1. *The DAG $\bar{G} = (V, A)$ can be reduced in polynomial time to a DAG G of size $3n \times 3n$.*

Proof. We will construct the DAG G by generating at most $3n$ columns and $3n$ rows. Consider the endpoints of the n slanted line segments given by (x_i, y_i) and $(x_i + 1, y_i + 1)$. Order the segments in increasing x -order in $O(n \log n)$ time. For each $1 \leq i \leq n$, add columns labeled $\lfloor x_i \rfloor$, $\lfloor x_i \rfloor + 1$, and $\lfloor x_i \rfloor + 2$. Repeat this process on the y coordinate, adding at most $3n$ rows, and yielding a grid of size at most $3n \times 3n$. Note that no slanted lines intersected any of the removed arcs and therefore any curve in the reduced grid can be mapped to a curve in the original grid intersecting the same number of slanted segments. \square

Finally, we use the longest path algorithm on G to find a curve C over the grid lines intersecting the most slanted segments. This route is feasible for the repairman and can be computed in $O(n^2)$ time by using dynamic programming [4, pp. 661-666].

In Figure 1 we show an example of the reduced DAG G with the slanted task segments and associated arc weights (left) as well as the solution corresponding to the longest path on G .

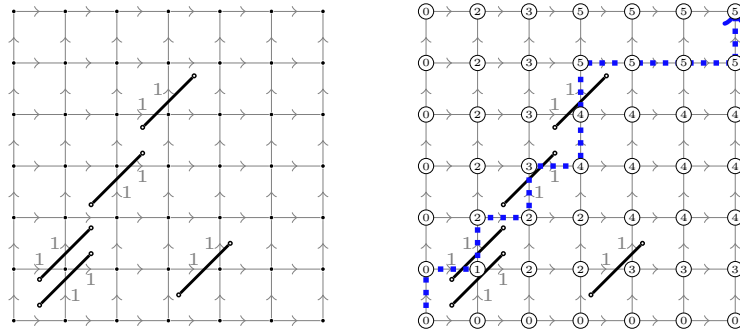


Fig. 1. Example of a reduced DAG G with the slanted segments and arc weights (left). The monotone curve attaining maximum weight on G (right).

Theorem 2 (Bar-Yehuda, Even, Shahar). *The described algorithm has approximation factor 8.*

The proof of Theorem 2 can be found in [1]

4 3-approximation Algorithm for Line-TRP-UTW

The algorithm described in Section 3 sometimes counts slanted segments twice. For example the solution of the example given in 1 gives a route of weight 5 but only intersects 3 slanted segments. S. L. Pérez Pérez *et al.* [3] proposed a modification to the algorithm which eliminates double counting, which we describe briefly.

We take the reduced DAG G and build an auxiliary DAG G' as follows:

1. We create a vertex for each horizontal arc in G and denote the set of all such vertices as V_h . Similarly, we obtain V_v from the vertical arcs in G .
2. We add an arc joining two vertices in V_h rightwards if the corresponding arcs were adjacent in G and call the set of all such arcs A_h . Analogously, we obtain A_v from joining vertices in V_v .
3. We add an arc from a vertex in V_h to a vertex in V_v if the corresponding arcs were adjacent in G and call the set of these arcs A_{hv} . Likewise, we obtain A_{vh} from arcs from a vertex in V_v to a vertex in V_h .

Then $G' = (V_h \cup V_v, A_h \cup A_v \cup A_{hv} \cup A_{vh})$. A section of G' is shown in Figure 2.

Lemma 1. G' can be constructed in $O(n^2)$ time.

Proof. By Theorem 1, the grid G has at most $9n^2$ vertices and every vertex has out-degree 2. Therefore, $|V(G')| = |A(G)| = 18n^2$. Every vertex in $V(G')$ also has out-degree at most 2, so $A(G') \leq 36n^2$. Assuming the creation of vertices and edges take constant time, constructing G takes $O(n^2)$ steps. \square

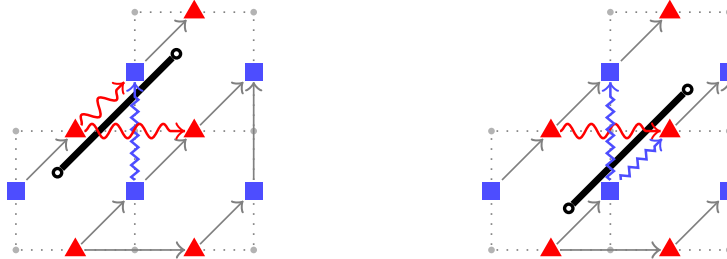


Fig. 2. A section of G' : vertices of V_h are triangles and vertices of V_v are squares. The edges that get weight 1 when a double counting slanted segment intersects the original grid are highlighted.

The new algorithm then becomes:

1. We create the auxiliary DAG G' in $O(n^2)$ time.
2. We assign the appropriate weights to each arc in $A(G')$ in $O(n^2)$ time.
3. We compute the longest path in G' by dynamic programming in $O(|V(G') + A(G')|) \in O(n^2)$ time.

In [3], S. L. Pérez Pérez *et al.* showed that this algorithm avoids the problem with double counting and has approximation factor 4. Its running time is $O(n^2)$.

4.1 Our Analysis of the New Algorithm

We start by giving an alternate proof that the new algorithm has approximation factor 4.

Lemma 2. *The new algorithm has approximation factor 4.*

Proof. Let P be the optimal route and let z_{OPT} be its cost. P can be covered by horizontal and vertical blocks on the grid as shown in Figure 4.1. Consider the four periodic routes r_1, r_2, r_3, r_4 show in Figure 4.1, with costs z_1, z_2, z_3, z_4 respectively, and the solution of the new algorithm A with cost z_A . Note that r_1, r_2, r_3, r_4 are all feasible paths for G' and together they intersect all edges intersected by P at least once. Therefore, $z_A \geq \max\{z_1, z_2, z_3, z_4\} \geq \frac{z_1+z_2+z_3+z_4}{4} \geq \frac{z_{\text{OPT}}}{4}$ \square

We use a similar idea to show that the approximation factor is at most 3.

Lemma 3. *The new algorithm has approximation factor 3.*

Proof. Let P be the optimal route and let z_{OPT} be its cost. P can be covered by horizontal and vertical blocks on the grid as shown in Figure 4.1. Consider the three periodic routes r_1, r_2, r_3 show in Figure 4.1, with costs z_1, z_2, z_3 respectively, and the solution of the new algorithm A with cost z_A . Note that r_1, r_2, r_3 are all feasible paths for G' and together they intersect all edges intersected by P at least once. Therefore, $z_A \geq \max\{z_1, z_2, z_3\} \geq \frac{z_1+z_2+z_3}{3} \geq \frac{z_{\text{OPT}}}{3}$ \square

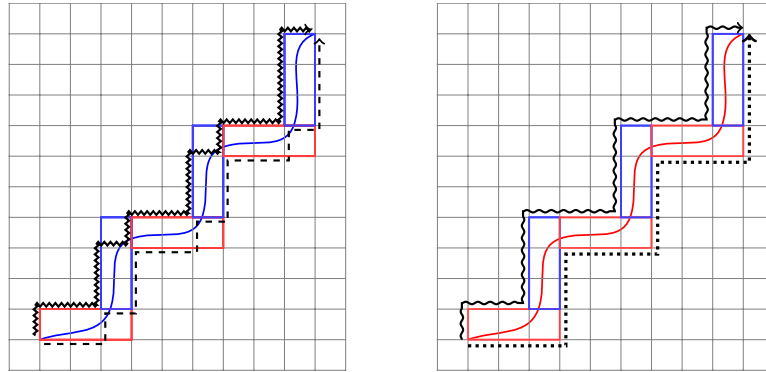


Fig. 3. Block cover of an optimal solution and the four periodic routes.

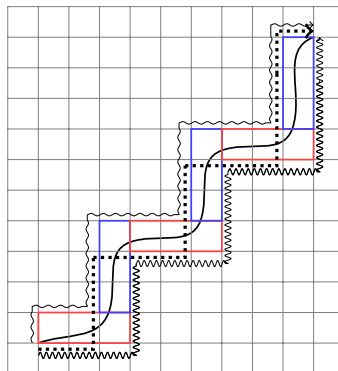


Fig. 4. Block cover of an optimal solution and the three periodic routes.

References

1. Bar-Yehuda, R., Even, G., Shahaar, S.M.: On approximating a geometric prize-collecting traveling salesman problem with time windows. *Journal of Algorithms* 55(1), 76–92 (2005)
2. Frederickson, G.N., Wittman, B.: Approximation algorithms for the traveling repairman and speeding deliveryman problems. *Algorithmica* 62(3-4), 1198–1221 (2012)
3. Pérez Pérez, S.L., Urbán Rivero, L.E., López Bracho, R., Zaragoza Martínez, F.J.: A fast 4-approximation algorithm for the traveling repairman problem on a line. In: *Proceedings of CCE*. pp. 268–271 (2014)
4. Sedgewick, R., Wayne, K.: *Algorithms*. Pearson Education (2011)
5. Tsitsiklis, J.N.: Special cases of traveling salesman and repairman problems with time windows. *Networks* 22(3), 263–282 (1992)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2015
Printing 500 / Edición 500 ejemplares

