Advances in Image Processing and Computer Vision

Research in Computing Science

Series Editorial Board

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 115,** septiembre 2016. 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. 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 responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 115**, September 2016. Printing 500. The authors are responsible for the contents of their articles. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 115

Advances in Image Processing and Computer Vision

Miguel González Mendoza (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2016

ISSN: 1870-4069

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

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Editorial

El propósito de este volumen es reflejar las nuevas direcciones de investigación y aplicaciones de los métodos de la Inteligencia Artificial en los campos de visión computacional y reconocimiento de patrones. Los artículos de este volumen fueron seleccionados con base en un estricto proceso de revisión efectuada por los miembros del comité de revisión, tomando en cuenta la originalidad, aportación y calidad técnica de los mismos. Cada artículo fue revisado por lo menos por dos miembros del comité de revisión del volumen (los revisores). Este volumen contiene 16 artículos relacionados con varios aspectos del desarrollo de los métodos de Inteligencia Artificial y ejemplos de sus aplicaciones a varias tareas tales como:

- Preprocesamiento de imágenes,
- Estegoanálisis de imágenes,
- Recuperación de imágenes por contenido,
- Extracción de características,
- Representación del conocimiento,
- Redes neuronales,
- Aprendizaje incremental,
- Computación evolutiva.

Este volumen puede ser interesante para los investigadores y estudiantes de las ciencias de la computación, especialmente en áreas relacionadas con la inteligencia artificial y su aplicación a los diferentes ámbitos de la vida cotidiana; así como, para el público en general interesado en estos fascinantes temas. En este número especial de la revista RCS, a nombre de la comunidad del Instituto Nacional de Óptica y Electrónica, en Puebla, Puebla, por apoyar de manera ingente la investigación, y el desarrollo de la ciencia y la tecnología, sustentado todo ello en un humanismo que transforma.

El proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible EasyChair, www.EasyChair.org.

Miguel Gonzalez-Mendoza

Editor invitado
ITESM, México

Septiembre 2016

Table of Contents

Page

Embellecimiento facial evolutivo dirigido por fuerzas

Ricardo Solano Monje[1], Nayeli Joaquinita Meléndez Acosta[2], Homero Vladimir Ríos Figueroa[3]

[1] Instituto Tecnológico Superior de Venustiano Carranza, Lázaro Cárdenas, Pue. México

[2] Universidad del Istmo, Campus Ixtepec, Ixtepec, Oax., México

[3] Universidad Veracruzana, Xalapa, Ver., México

rsolano@itsvc.edu.mx, nayelimelendez@gmail.com, hrios@uv.com.mx

Resumen. Este trabajo explica el desarrollo de una propuesta original para embellecer rostros, resultado de un algoritmo evolutivo y un algoritmo de dibujo estético de grafos. El algoritmo de dibujo estético de grafos opera en este contexto como el operador de mutación. La estrategia dirigida por fuerzas que es el fundamento del algoritmo de dibujo estético de grafos, hace uso de las leyes de Hooke que gobiernan la tensión en resortes. La solución de los resortes utiliza un modelo mecánico para producir diseños 2D "estéticamente agradables", calculando fuerzas repulsivas y atractivas entre nodos. En nuestro caso, los nodos que describen un grafo son una representación del rostro a embellecer. El objetivo de este trabajo es crear un rostro embellecido en comparación con el rostro de entrada. Un individuo *I* ésta compuesto por la máscara del rostro de entrada, es decir Puntos de Referencia y Vector de Distancias. En los resultados experimentales se utilizó la base de datos de imágenes Fg-Net [7]. Nuestros resultados experimentales muestran rostros embellecidos usando nuestra aproximación, mostrando así una solución competitiva.

Palabras clave: Trazado estético de grafos, embellecimiento de rostros, representación de rostros por apariencia, eigenspaces de rostros.

Force-directed Evolutionary Face Beautification

Abstract. This work explains the development of a novel approach to beautify a face, merging an evolutionary approach and aesthetics graph drawing. The objective of this work is to build a beautify face seen against the input face. The aesthetics graph drawing algorithm works as the mutation operator. The approach of force-directed graph drawing which in turn is the solution to aesthetical graph drawing makes use of the law of Hooke about springs. The springs embedded approach uses a mechanical model to build 2D layouts aesthetically pleasing by computing attraction and repulsion forces in between nodes. In our case of face beautification, the graph nodes count as a representation of the face to be beautified. We have carried out tests using the Fg-Net face database [7]. Our

experimental results show beautified faces using our approach, meaning our approach as a competitive solution.

Keywords: Aesthetically pleasing graph layouts, face beautification, appearance representation of the face, modular eigenspaces of the face.

1. Introducción

Es perfectamente sano asumir que cualquier modelo que encontramos hoy en día en la portada de una revista ha sido manipulada por un especialista con habilidades de retoque. El desarrollo de una herramienta inteligente de embellecimiento tendrá impactos en cine y propaganda, entre muchos otros.

El embellecimiento facial por computadora es un tópico de Visión por Computadora. Este trabajo de investigación se centra únicamente en el aspecto de embellecer el rostro.

El contenido del artículo está organizado en 6 secciones de la siguiente manera: en la segunda sección se muestran algunos trabajos relacionados al problema del embellecimiento artificial. La tercera sección muestra el marco teórico del algoritmo de dibujo estético de grafos dirigido por fuerzas, utilizado como operador de mutación en el Algoritmo Evolutivo. En la cuarta sección se explica el funcionamiento del Algoritmo Evolutivo. La quinta sección muestra las pruebas y resultados obtenidos. Finalmente la sexta sección corresponde a las conclusiones.

2. Embellecimiento del rostro

El reconocimiento de rostro se lleva a cabo con información de bulto, por ello los detalles pueden ser perdonados o compensados cuando se ve el rostro por segunda vez y éste no es tan bello como la primera vez —la vez que se vio embellecido—. Es decir, que por analogía al reconocimiento de rostros, el embellecimiento artificial pasaría desapercibido hasta cierto punto.

El embellecimiento digital del rostros toma de entrada una fotografía —imagen frontal de un rostro y de forma automática mejora el atractivo del rostro e incrementa la calificación de belleza (la calificación predicha por la máquina de belleza, que califica el embellecimiento) del rostro.

Uno de los principales objetivos es lograr un rostro embellecido que mantenga una semejanza marcada —no dudosa— con el rostro original en una imagen de entrada por medio de algoritmos evolutivos.

En [3] Leyvand realiza la modificación y embellecimiento del rostro creando una máquina embellecedora construyendo una máscara con 84 puntos de referencia, con esta mascara se forma un vector de distancias entre los puntos de referencia que definen las 8 características faciales: las dos cejas, los dos ojos, el labio inferior y superior de la boca, la nariz y límites del rostro.

Eisenthal en [4] persigue en su trabajo calificar la belleza, es decir construye una máquina calificadora y solo localiza 37 puntos de referencia utilizando una SVM

(Support Vector Machine), la cual es utiliza por Leyvand para calificar la belleza del rostro.

En [5] Solano realiza el embellecimiento del rostro y en [6] realizó una máquina calificadora, construye una máscara con 66 puntos de referencia. Esta mascara ha sido tomada como referencia en este trabajo. La Fig. 1(a) muestra los 66 puntos de referencia. La Fig. 1(b) las características faciales y la Fig. 1(c) la máscara formada por 89 distancias.

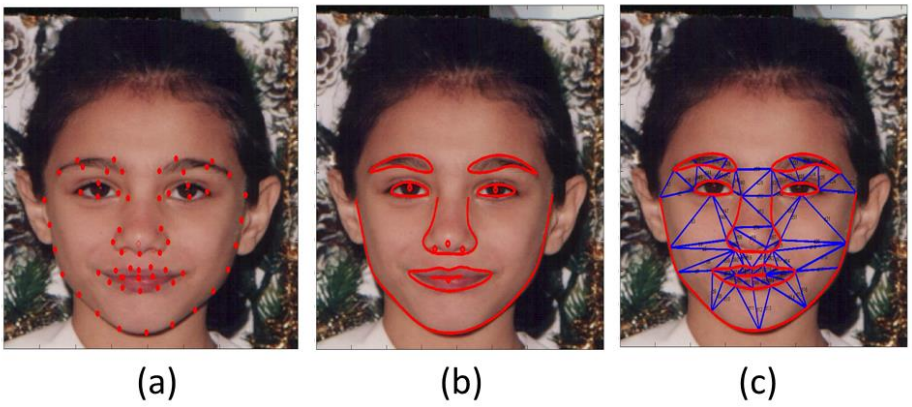


Fig. 1. (a) Los 66 puntos de referencia. (b) Las características faciales. (c) Las 89 distancias entre puntos de referencia

3. Algoritmo de dibujo estético de grafos dirigido por fuerzas

Un grafo $G = (V, E)$ es un par dónde V es el conjunto de vértices y E es el conjunto de aristas. Un dibujo de un grafo G en el plano es un mapeo D de V a R, dónde R es el conjunto de números reales. Es decir, cada vértice v se coloca en el punto $D(v)$ en el plano, y cada artista (u, v) se muestra como un segmento de línea recta que conecta $D(u)$ y $D(v)$.

Los métodos de Fuerza dirigida construyen dibujos en línea recta de grafos generales utilizando un modelo físico en el que los vértices y las aristas del grafo son vistos como objetos físicos sujetos a varias fuerzas.

En 1984 el algoritmo de Eades [6] fue hecho para grafos con un máximo de 30 vértices y utiliza un modelo mecánico para producir diseños 2D "estéticamente agradables". El algoritmo se resume brevemente como sigue:

En esta versión, la fuerza resultante en cada nodo son la suma de las fuerzas atractivas (utilizando los nodos vecinos con arista adyacente) y de las repulsivas (calculada utilizando los nodos sin arista). Las fuerzas atractivas (entre nodos que tiene aristas) tienen la forma:

$$f_a(u, v) = c_1 \times \log\left(\frac{l}{c_2}\right), \tag{1}$$

donde c_1 y c_2 son constantes y l es la longitud de la arista.

En tanto las fuerzas repulsivas (entre los nodos que no tienen aristas) quedan determinadas por la siguiente fórmula:

$$f_r(u,v) = \left(\frac{c_3}{l^2}\right), \tag{2}$$

donde c_3 es una constante y l es la distancia entre nodos.

La fuerza $f(u)$ que experimenta un vértice u es

$$f(u) = \sum_{(u,v)\in E} f_a(u,v) + \sum_{(u,v)\notin E} f_r(u,v). \tag{3}$$

Algoritmo 1. Dibujo estético de grafos dirigido por fuerzas propuesto por Peter Eades (1984) *A Heuristic for Graph Drawin.*

Algoritmo (G: grafo);

Colocar los vértices de G en localizaciones aleatorias
Repetir M veces
Calcular la fuerza en cada vértice;
Mover el vértice c_4 * (Fuerza en el vértice);
Dibujar grafo;

Los valores c_1 = 2, c_2 = 1, c_3 = 1 y c_4 = 0.1, son apropiados para la mayoría de los grafos. La mayoría de los grafos alcanzan un estado de energía mínima después de 100 veces, es decir, M = 100 [6].

Las Fig. 2 y 3 muestran la representación del algoritmo de dibujo estético de grafos dirigido por fuerzas enfocado al problema de esta investigación, embellecimiento digital del rostro por partes. En la Fig. 2 se observa la Boca modelada a través de resortes y en la Fig. 3 se observan las cejas modelas con resortes.

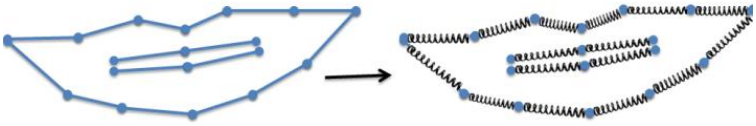


Fig. 2. La característica facial "La Boca", representación con resortes

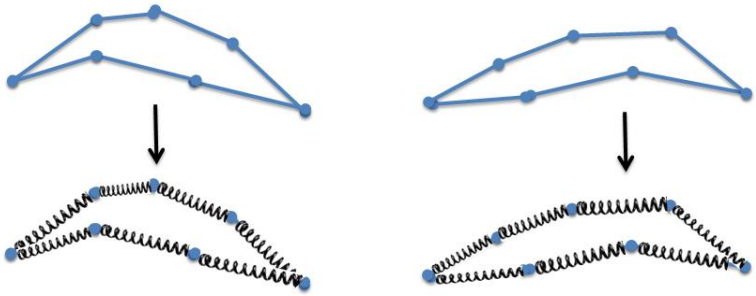


Fig. 3. La característica facial "Las Cejas", representación con resortes

4. Embellecimiento evolutivo usando el algoritmo de dibujo estético de grafos dirigido por fuerzas

Esta investigación describe el desarrollo de un algoritmo de embellecimiento facial, que fusiona un algoritmo de trazado estético de grafos con un algoritmo evolutivo. En particular, el operador de mutación del algoritmo evolutivo es implementado a partir de trazo estético dirigido por fuerzas. Con la unión de ambos algoritmos se busca generar un rostro más bello que el rostro de entrada.

Este trabajo toma la idea de enfocar utilizada en [5], esto significa que sólo una característica facial es tomada para realizar el proceso de embellecimiento y el método se concentra únicamente sobre esta. El sistema recibe de entrada una imagen frontal del rostro, junto con sus respectivos puntos de referencia. Los puntos de referencia de la característica seleccionada son llamados puntos seleccionados, el resto son puntos fijos. La Fig. 4(a) muestra el caso de seleccionar la boca como característica a embellecer y en la Fig. 4(b) han sido seleccionadas las cejas para su embellecimiento.

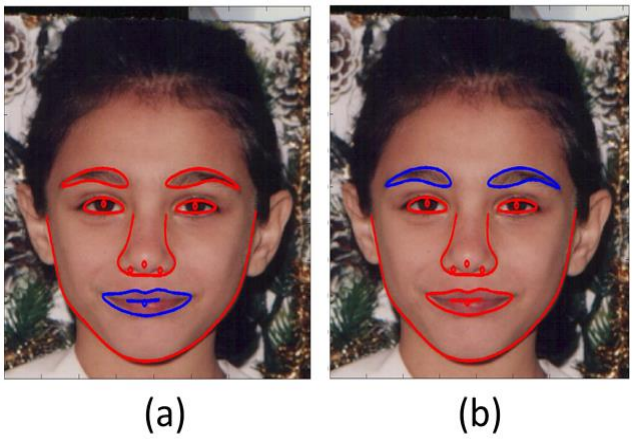


Fig. 4. Selección de la Característica a Embellecer: a) la boca y b) las cejas

El proceso de embellecimiento se muestra en la Fig. 5. Dada una foto frontal como entrada el usuario selecciona la característica facial a embellecer. A partir de los puntos de referencia, se extrae un vector de distancias, que corresponden a la máscara del rostro de entrada. Esta mascara pasa al Algoritmo Genético, el cual proporciona como resultado un vector de distancias modificado que posee una calificación de belleza más alta que el vector original.

La transformación warping es utilizada para modificar la imagen de entrada, este proceso mapea el conjunto de puntos de referencia $\{pi\}$ de la imagen origen en el correspondiente conjunto de puntos de referencia $\{qi\}$ de destino.

4.1. Funcionamiento del algoritmo evolutivo

Un individuo I ésta compuesto por la máscara: Puntos de Referencia y Vector de Distancias. El funcionamiento principal (o ciclo evolutivo) del Algoritmo Evolutivo se realiza de la siguiente forma:

1. Generar una población inicial de 21 individuos formada por el individuo original I y 20 versiones mutadas generadas utilizando el algoritmo de dibujo estético de grafos dirigido por fuerzas.
2. Evaluar a la población actual.
3. El mejor individuo (es aquel que tiene la mejor calificación proporcionada por la función de evaluación) es seleccionado para ser el primer individuo de la nueva población.
4. Seleccionar 10 padres de la población actual por *certamen*.
5. Cruzar a los padres seleccionados, los descendientes se incorporan a la nueva población.
6. Utilizar al mejor individuo de la población para generar 10 descendientes mutados haciendo uso del algoritmo de dibujo estético de grafos dirigido por fuerzas. Estos complementan los 21 individuos de la nueva población.
7. Reemplazar la población actual por la nueva población.
8. Volvemos al paso hasta completar el número de generaciones.

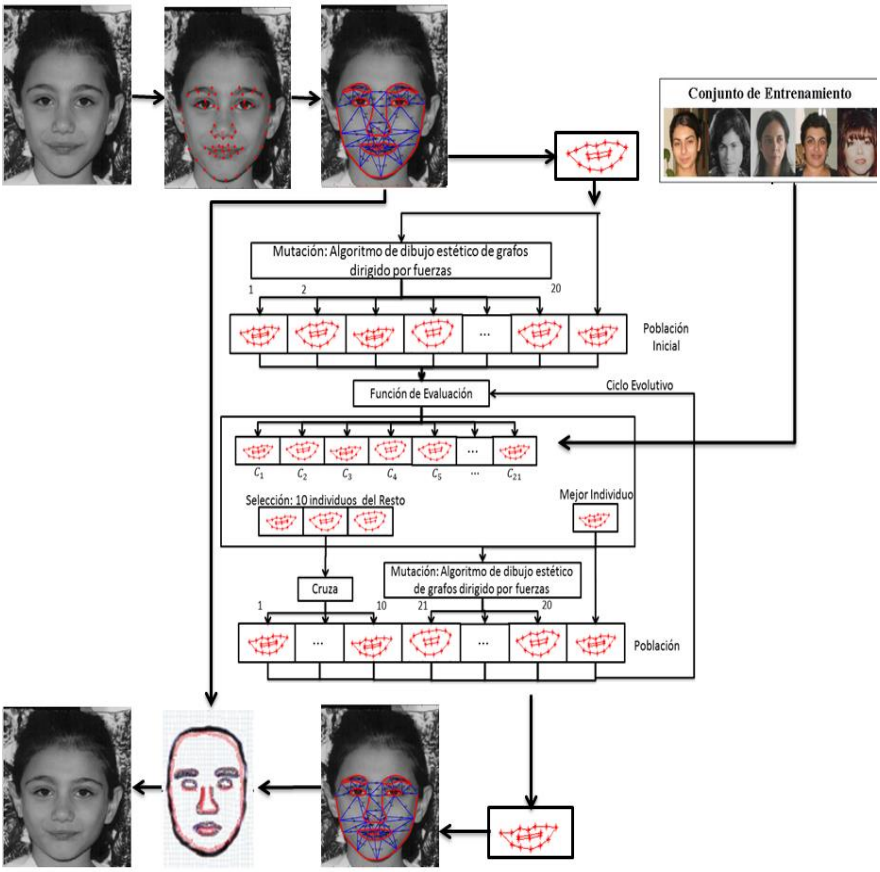


Fig. 5. Procedimiento de Embellecimiento la BOCA como característica a embellecer

4.2. Función de aptitud

La función de evaluación utilizada para evaluar la belleza en esta investigación hace uso de eigenespacios (usados para calificar la belleza en [4], usados en otros trabajos [5,6,7,8,9]) los cuales tomando como ensamble vectores de puntos antropométricos de rostros bellos. El Algoritmo Evolutivo genera un vector vecino v' con una calificación de belleza más alta $(v') > (v)$, donde v es el vector original.

4.3. Selección

La selección de la población para cruce se realiza por *certamen*: 20 individuos participan en competencias *partwise*, es decir para cualquier I_1 e I_2 (elegidos aleatoriamente) que aún no han participado en una competencia, si $(I_1) > (I_2)$ es seleccionado el Individuo I_1, en caso contrario es seleccionado el Individuo I_2.

4.4. Cruce y mutación

Los individuos seleccionados (o padres) son cruzados utilizando el método cruce en un punto. Después a partir del mejor individuo mediante mutación se generan 10 individuos utilizando el algoritmo de dibujo estético de grafos dirigido por fuerzas, en este algoritmo es muy importante la determinación de cuatro constantes. c_1, c_2, c_3 y c_4, para poder generar individuos válidos.

5. Pruebas y resultados

Los parámetros c_1, c_2, c_3 y c_4 utilizados en el algoritmo de dibujo estético de grafos dirigido por fuerzas son ajustados experimentalmente de acuerdo a sus efectos sobre la visualización de los grafos que forman cada característica del rostro. Por lo anterior se realizaron tres experimentos considerando los parámetros c_1 y c_4, que nos permitió ajustar el intervalo de valores que pueden tomar estos.

Los experimentos realizados muestran que los ajustes a los parámetros permiten la generación de grafos validos los cuales se visualizaran de una forma mucho más clara. También en los experimentos identificamos que solo es necesario realizar el ajuste en los parámetros c_1 y c_4, esto se atribuye a que el parámetro c_1 es usado en el cálculo de las fuerzas atractivas (entre el conjunto de nodos que tiene aristas) y el parámetro c_4 determina la tasa de cambio repulsivo o atractivo. Dichos experimentos se definen a continuación:

Experimento 1: La Fig. 6 muestra seis resultados, tres *variando el parámetro* c_1*:* (a) c_1=1, (b) c_1= 0.5, (c) c_1=0.125, dos *variando el parámetro* c_4*:* incisos (d) c_4= 0.0125 y (e) c_4= 0.0625. Por último un experimento *variando los parámetros* c_1 *y* c_4*:* es la combinación de variar ambas parámetros (f) c_1=0.125 y c_4=0.0125.

En este experimento podemos observar que conforme la iteración incrementa el grafo se deforma en este caso para nuestro problema. Es decir si dejamos constante este parámetro al final se producirían grafos no útiles para nuestro problema.

Experimento 2: En un experimento 2 se variaron los parámetros c_1 y c_4 aleatoriamente, la Fig. 7 muestra algunos resultados variando aleatoriamente los parámetros *utilizando el mismo valor del parámetro en cada una de las aristas que conforman el grado de la característica facial*.

Este experimento muestra que podemos obtener variabilidad muy rápido debido a la aleatoriedad, pero aun así se puede observar que el grafo solo crece.

Experimento 3: Por último se realizó un tercer experimento en el cual se variaron los parámetros c_1 y c_4 aleatoriamente *en cada una de las aristas que conforman la el grafo de la característica facial*. La Fig. 8 muestra algunos resultados.

Podemos observar en los resultados que la aleatoriedad en cada arista nos permite tener grafos con diferentes formas más variables.

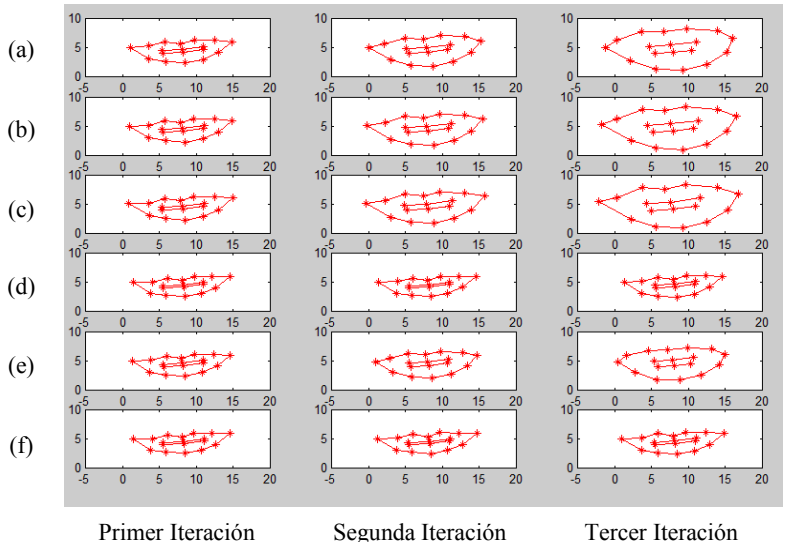


Fig. 6. Experimento 1: Las bocas generadas variando los parámetros c_1 *y* c_4. (a) c_1=1, (b) c_1= 0.5, (c) c_1=0.125, (d) c_4= 0.0125, (e) c_4= 0.0625 y (f) c_1=0.125 y c_4=0.0125

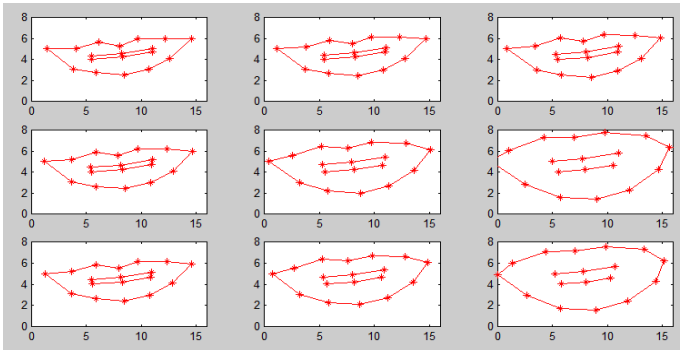


Fig. 7. Experimento 2: Bocas generadas variando los parámetros c_1 y c_4 aleatoriamente, pero utilizando estos mismos parámetros en cada una de las aristas que conforman la boca

Basados en los experimentos y resultados se determinaron los rangos de valores que pueden tomar los parámetros: $c_1 \in [0.0625, 2]$ y $c_4 \in [0.00625$ y $0.1]$.

Algunos resultados de embellecimiento evolutivo para diferentes características faciales, utilizando el algoritmo de dibujo estético de grafos dirigido por fuerzas como operador de mutación se observan en la Fig. 9, 10, 11, 12, 13,14.

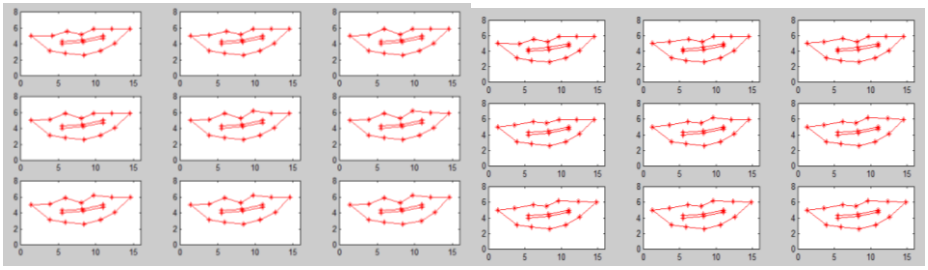


Fig. 8. Bocas generadas variando los parámetros c_1 y c_4 aleatoriamente, utilizando diferentes valores del parámetro en cada una de las aristas que conforman la boca

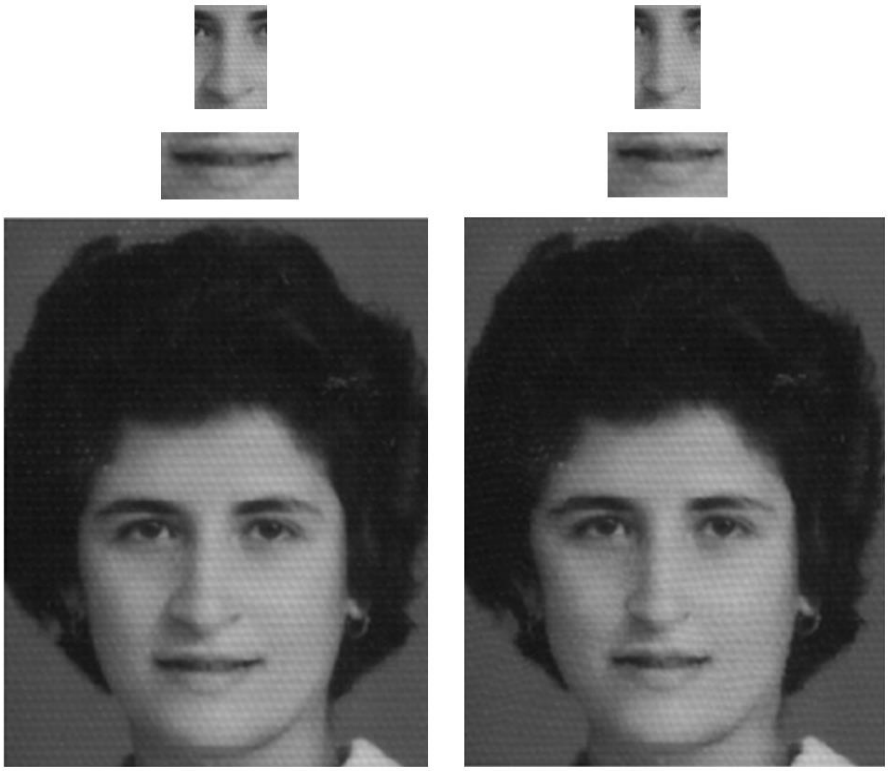


Fig. 9. Resultados de embellecimiento. A empezado a evolucionar las características faciales: "BOCA, CEJAS, NARIZ Y LIMITES DEL ROSTRO". Número de Generaciones 100. Izquierda: rostro original y Derecha: rostro embellecido

Fig. 10. Resultados de embellecimiento. A empezado a evolucionar las características faciales: "CEJAS, BOCA y OJOS". Número de Generaciones 50. Izquierda: rostro original y Derecha: rostro embellecido

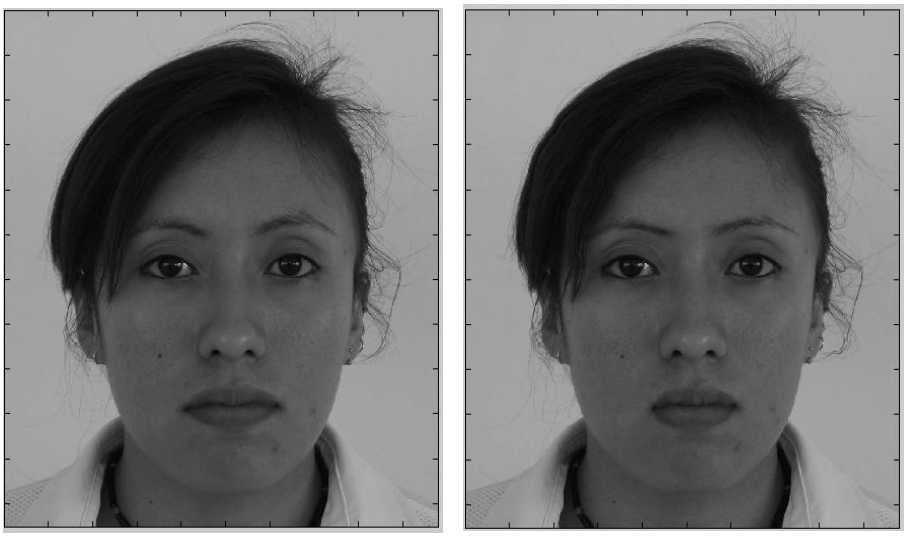


Fig. 11. Resultados de embellecimiento. A empezado a evolucionar las características faciales: "CEJAS, BOCA y LIMITES DEL ROSTRO". Número de Generaciones 50. Izquierda: rostro original y Derecha: rostro embellecido

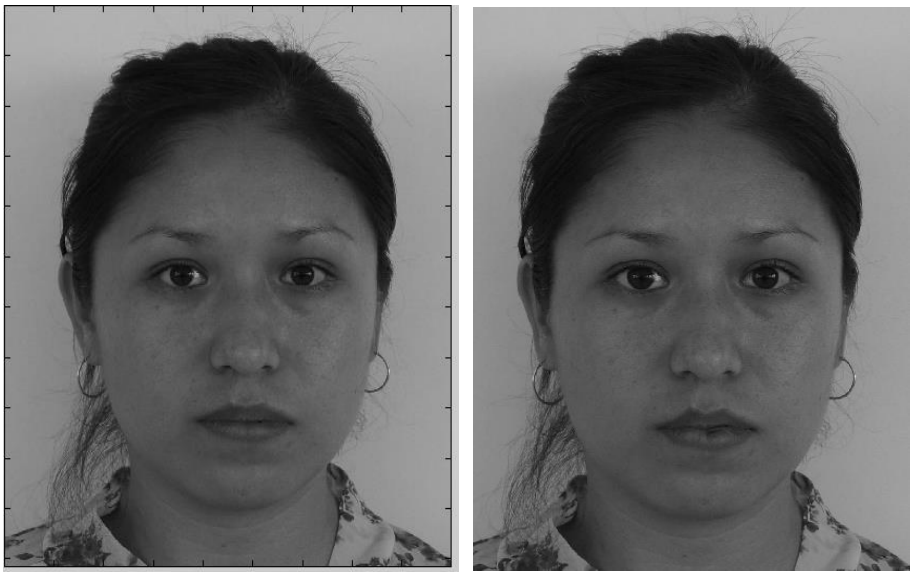


Fig. 12. Resultados de embellecimiento. A empezado a evolucionar las características faciales: "BOCA, CEJAS, BOCA y LIMITES DEL ROSTRO". Número de Generaciones 50. Izquierda: rostro original y Derecha: rostro embellecido

Fig. 13. Resultados de embellecimiento A empezado a evolucionar la característica facial: "NARIZ". Número de Generaciones 50. Izquierda: rostro original y Derecha: rostro embellecido

Para probar el funcionamiento de nuestro sistema se realizó un encuesta a estudiantes de licenciatura a los cuales se les mostro el rostro original junto con el mejorado y ellos indicaron cual es el rostro más bello. Se han considerado 40

estudiantes para la encuesta, en promedio el 82% de los estudiantes acertaron cual es la imagen embellecida.

Fig. 14. Resultados de embellecimiento A empezado a evolucionar las características faciales: "BOCA, CEJA y LIMITES DEL ROSTRO". Número de Generaciones 50. Izquierda: rostro original y Derecha: rostro embellecido

6. Conclusiones

Esta investigación describe el desarrollo de un algoritmo de embellecimiento facial, que fusiona un algoritmo de trazado estético de grafos con un algoritmo evolutivo. En particular, el operador de mutación del algoritmo evolutivo es implementado a partir de trazo estético dirigido por fuerzas. Con la unión de ambos algoritmos se busca generar un rostro más bello que el rostro de entrada.

El sistema está limitado a recibir una imagen frontal del rosto junto con sus respectivos puntos de referencia.

Los experimentos realizados permiten terminan los intervalos de valores para los parámetros: $c_1 \in [0.0625, 2]$ y $c_4 \in [0.00625$ y $0.1]$.

Referencias

1. Leyvand, T., Cohen, Or D., Dror G., Lischinski D.: Data Driven Enhancement of Facial Attractiveness. ACM Trans. Graph, Vol. 27, No. 3, p. 9 (2008)
2. Eisenthal, Y., Dror, G., Ruppin, E.: Facial attractiveness: Beauty and the machine. Neural Computation, Vol. 18, No. 1, pp. 119–142 (2006)

3. Solano Monje, R., Meléndez Acosta, N.J., Juárez Vázquez, S., Rios Figueroa, H.V.: Belleza Artificial: Evolucionando partes del Rostro. Research in Computing Science, Vol. 93, pp. 111-120 (2015)
4. Solano Monje, R., Meléndez Acosta, N.J., Juárez Vázquez, S., Rios Figueroa, H.V.: Eigenespacios de Belleza Paramétricos como Máquina Calificadora. Research in Computing Science, Vol. 93, pp. 133–140 (2015)
5. Murase, H., Nayar, S.K.: Parametric Eigenspace Representation for Visual Learning and Recognition. In: Proceedings of The International Society for Optical Engineering (SPIE), Vol. 2031, pp. 378–391 (1993)
6. Nayar, S.K., Murase, H., Nene, S.A.: Parametric Appearance Representation. In: Early Visual Learning, pp. 131–160 (1996)
7. Nayar, S.K., Baker, S., Murase, H.: Parametric Feature Detection. In: DARPA Image Understanding Workshop (IUW), pp. 1425–1430 (1997)
8. Pentland, A., Moghaddam, B., Starner, T.: View-Based and Modular Eigenspaces for Face Recognition. In: IEEE Conference on Computer Vision & Pattern Recognition (1994)
9. Solano Monje, R.: Modelado y reconocimiento de objetos usando apariencia y multi-eigenespacios particionados. Instituto Nacional de Astrofísica, Óptica y Electrónica, Septiembre (2002)
10. Kobourov, S.G.: Unit 12: Force-Directed Drawing Algorithms. University of Arizona. (2004)
11. Utech, J., Branke, J., Schmeck, H., Eades, P.: An Evolutionary Algorithm for Drawing Directed Graphs. In: Internacional Conference on Imaging Science, Systems, & Technology (CISST'98), pp. 154–160 (1998)
12. Cootes, T.: The Fg-Net Aging Database, http://sting.cycollege.ac.cy/~alanitis/ fgnetaging/index.htm

Clasificación de la manzana royal gala usando visión artificial y redes neuronales artificiales

Gustavo Andrés Figueredo Avila

Universidad Pedagógica y Tecnológica de Colombia,
Escuela de Ingeniería de Sistemas y Computación,
Tunja, Boyacá,
Colombia

gustavoandres.figueredo@uptc.edu.co

Resumen. La clasificación de una fruta según su estado de maduración es complicada debido a todas las variables que influyen en ese cambio. Se propuso la construcción de una red neuronal artificial multicapa backpropagation (BPNN) con aprendizaje supervisado entrenada con cuatro diferentes algoritmos de aprendizaje (TRAINSCG, TRAINBFG, TRAINBR, TRAINLM) de manera independiente. De un conjunto de 30 manzanas Royal Gala, se adquirieron un total de 4200 fotos durante 35 días, que fueron procesadas y redimensionadas a un tamaño de 256x456. Las fotos segmentadas se convirtieron a modelo de color RGB, HSV y Lab, con el fin de extraer los valores de la media, la varianza y la desviación estándar. El mejor rendimiento de entrenamiento se obtuvo con el algoritmo TRAINBR (99.3%), y el mejor rendimiento en pruebas se obtuvo con TRAINSCG (67.7%). Finalmente, las comparaciones entre los diferentes resultados de cada algoritmo muestran que el mejor algoritmo de clasificación fue TRAINSCG.

Palabras clave: Procesamiento de imágenes digitales, red neuronal artificial Backpropagation (BPNN), Red Neuronal Artificial (RNA), algoritmo de entrenamiento, Mean Squared Error (MSE).

Classification of Royal Gala Apple Using Computer Vision and Artificial Neural Networks

Abstract. The fruit classification according to their state of maturation is complicated because of all variables that influence this change. In this paper a backpropagation multilayer artificial neural network (BPNN) is proposed, it was trained with supervised learning with four different learning algorithms (TRAINSCG, TRAINBFG, TRAINBR, TRAINLM) independently. During 35 days, a total of 4200 pictures were acquired from a set of 30 Royal Gala apples, which were processed and resized to a size of 256x456. The segmented pictures were converted to RGB, HSV and Lab color models, in order to extract the values of the mean, variance and standard deviation. The best training performance was obtained with the algorithm TRAINBR (99.3%) and the best test performance was obtained with TRAINSCG (67.7%). Finally, comparisons between different results from each algorithm show that the best classification algorithm was TRAINSCG.

Keywords: Digital image processing, Backpropagation Artificial Neural Network (BPNN), Artificial Neural Network (ANN), training algorithm, Mean Squared Error (MSE).

1. Introducción

La identificación del comportamiento y cambio constante del proceso de maduración de una fruta durante un periodo determinado de tiempo, conlleva el análisis de diferentes variables que afectan dichos cambios y que de antemano se les atribuye un comportamiento impredecible o no lineal [1]. El análisis técnico de los principales cambios físicos de una fruta a lo largo de su vida útil, normalmente se realiza utilizando técnicas de espectrometría, que abarcan los cambios más significativos relativos al color; o análisis destructivo para medir las variaciones de líquidos, alcohol, almidón y azúcares [2].

Usualmente, la determinación del estado de madurez de una fruta se encuentra en manos de los humanos, que son capaces de entrar a considerar un número significativo de cambios en la estructura física de las frutas, y dar un resultado cualitativo respecto a la calidad, aunque al igual que las técnicas tecnológicas tradicionales, dicha clasificación suele ser provechosa en periodos de tiempo relativamente largos.

Las Redes Neuronales Artificiales (RNA) han sido principalmente usadas para la clasificación de datos, reconocimiento de patrones y predicción de comportamientos, debido a su intención de emular la función de las redes neuronales biológicas, a las que se les atribuye el proceso de aprendizaje mediante ejemplos [3, 4,5]. Para el caso de clasificación y reconocimiento de patrones en frutas, usualmente se ha venido usando la arquitectura de red neuronal denominada perceptrón multicapa backpropagation, la cual puede ser entrenada con diferentes algoritmos. En la literatura se menciona el uso de redes neuronales artificiales para clasificar manzanas según el color [6], para identificar diferentes tonalidades de color, magulladuras o daños en la superficie de la manzana [7], o para clasificar diferentes tipos de frutas según ciertas características extraídas de la imagen [8].

Para esta investigación se usó la Red Neuronal Backpropagation (BPNN), entrenada por separado con algoritmos Gradiente Conjugado Escalado (TRAINSCG), BFGS Quasi Newton (TRAINBFG), Regularización Bayesiana (TRAINBR) y Levenberg-Marquardt (TRAINLM), con una arquitectura de 3 capas, un total de 30 neuronas de entrada, 28 neuronas en la capa oculta y 5 neuronas de salida.

Esta investigación incluye la sección 2, en donde se describen los materiales y métodos utilizados para el experimento, la sección 3, en donde se especifican los pasos en el procesamiento de las imágenes, en la sección 4 se especifica la arquitectura de la red neuronal, en la sección 5, los resultados, y en la sección 6 las conclusiones y trabajos futuros.

2. Materiales y métodos

Para el desarrollo de esta investigación, se adquirieron un total de 30 manzanas Royal Gala recién llegadas al supermercado, las cuales fueron dispuestas diariamente sobre y delante una superficie blanca por un periodo de 35 días. A cada manzana se le

tomó una foto por cada una de las 4 caras de la manzana previamente marcadas, bajo la presencia de luz natural. Diariamente se obtuvieron un total de 240 fotografías.

Las fotografías fueron adquiridas en formato JPG (Grupo Conjunto de Expertos en Fotografía) bajo el modelo RGB (Red, Green, Blue – Rojo, Verde, Azul) con una dimensión de 2592x1456 y una resolución de 72ppp. No se aplicó flash ni zoom digital.

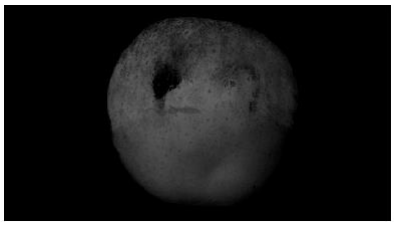
A cada manzana se le tomaron datos iniciales de peso, altura y diámetro ecuatorial, durante los primeros días del experimento, y posteriormente se retomaron en los últimos días, con el propósito de conocer la variación por cada día trascurrido.

3. Procesamiento de imagen

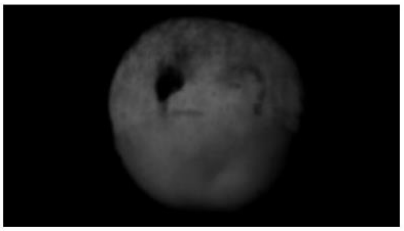
3.1 Pre-Procesamiento

Cada imagen adquirida (Ver Figura 1 (a)) fue inicialmente modificada eliminando los valores del canal verde y azul con el fin de obtener la forma de la manzana (Ver Figura 1 (b)). Posteriormente se le aplicó un filtro de Gauss que permitiera difuminar la imagen y eliminar la mayor cantidad de ruido presente sobre la misma, y a su vez suavizar los bordes (Ver Figura 1 (c)).

a) Imagen Original (2592x1456)



b) Imagen Canal Rojo menos Verde y Azul.

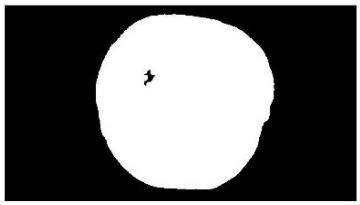


c) Imagen Filtro Gauss.

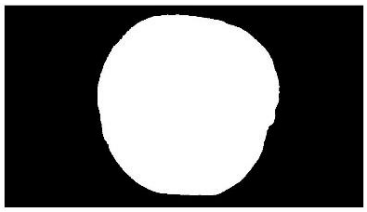
Fig. 1. Detección de forma y eliminación de ruido

Una vez obtenida la forma de la manzana, se realizó el proceso de binarización, con el cual se obtuvo una imagen representada únicamente en dos valores (0 y 1) (Ver Figura 2 (a)). En algunos casos la imagen binarizada presentó la presencia de 'Huecos', los cuales son bits no pertenecientes al conjunto de bits predominantes en un área, al ser eliminados se obtuvo la forma completa de la manzana (Ver Figura 2 (b)).

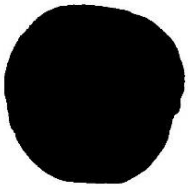
Paso siguiente, se invirtieron los valores de los bits de la imagen binarizada (Ver Figura (c)), para superponer la imagen original sobre la máscara binaria creada (Ver Figura 2 (d). La imagen final se redimensionó a 256x456 pixeles para reducir el tiempo de procesamiento en la obtención de datos.



(a) Binarización Imagen.



(b) Relleno de 'Huecos'



(c) Inversión de bits.

(d) Imagen Segmentada (256x456 pixeles).

Fig. 2. Creación de mascara binaria y superposición de imagen original

3.2 Obtención de datos

Las imágenes segmentadas y redimensionadas originalmente en modelo RGB fueron transformadas a los modelos de color HSV (Hue, Saturation, Value – Matiz, Saturación, Valor) y Lab (Luminosidad, Rojo-Verde, Gradiente Azul), para extraer la información de cada uno de los canales calculando la media, la varianza y la desviación estándar, generando de esta forma una matriz de 9x4200 por cada modelo de color, en donde el 9 representa la concatenación de los 3 valores estadísticos calculados por cada canal de color, y los 4200 datos son el resultado de 30 manzanas x 4 caras x 35 días.

La media se calcula mediante la ecuación 1, en donde P representa cada uno de los tres canales de cada modelo de color RGB, HSV, Lab:

$$\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i. \tag{1}$$

La varianza se calcula mediante la ecuación 2, en donde P representa cada uno de los tres canales de cada modelo de color RGB, HSV, Lab y $\bar{P}$ es la media calculada de cada canal:

$$\sigma_p{}^2 = \frac{1}{n}\sum_{i=1}^{n} (P_i - \bar{P})^2. \tag{2}$$

La desviación estándar se calcula mediante la ecuación 3:

$$\sigma_p = \sqrt{\sigma_p{}^2}. \quad (3)$$

Posteriormente se calculó el valor de la media de las cuatro caras que corresponden a cada manzana, con lo cual se redujo el tamaño de la matriz a 27 x 1050, que representan las 9 entradas de cada modelo de color por 30 manzanas durante un periodo de 35 días.

Los datos de entrada correspondientes a los tres modelos de color seleccionados, proporcionaron información sobre el cambio en la coloración, matiz, saturación e iluminación en la superficie de la manzana. Se complementaron los datos de entrada asociados al color, con los valores de peso, altura y diámetro ecuatorial, resultados de un cálculo interpolado basado en datos adquiridos en los primeros días del experimento y al final del experimento.

Finalmente se consolidaron los datos en una matriz de 30 x 1050 que se usa para el entrenamiento de la red neuronal.

4. Arquitectura de red neuronal

4.1 Red neuronal

La red neuronal artificial planteada consta de un total de tres capas (la capa de entrada, la capa oculta y la capa de salida). La capa de entrada está compuesta por 30 neuronas (representan la media, la varianza y la desviación estándar por cada canal (3 canales) de cada modelo de color (3 modelos)), la capa oculta por 28 neuronas y la capa de salida por 5 neuronas (Representan cinco semanas, cada una de siete días).

La red es entrenada inicialmente por una matriz de 30x1050 de los cuales el 75% de los datos se usaron como entrenamiento, el 15% como validación y el 15% como pruebas. Posteriormente se realizó otro entrenamiento únicamente con una matriz de 30x750, dejando una matriz de 30x350 como matriz de prueba independiente. La arquitectura de la red se puede apreciar en la Figura 3.

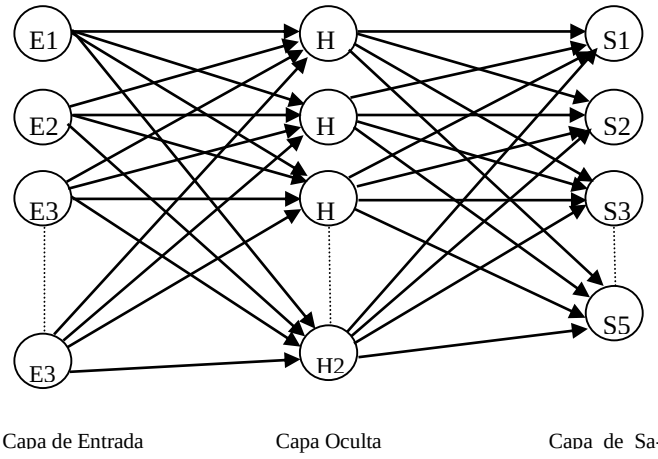


Fig. 3. Arquitectura Red Neuronal Multicapa

4.2 Algoritmos de entrenamiento

Los Algoritmos de entrenamiento modifican los pesos de cada neurona para identificar a lo largo de una serie de iteraciones, cual es el patrón que mejor se ajusta a los objetivos. Se calcularon los valores de rendimiento MSE (Mean Squared Error) para cada algoritmo, los cuales indican el error mínimo alcanzado para el entrenamiento o prueba.

En esta investigación se propuso utilizar como algoritmos de entrenamiento para la red neuronal backpropagation los siguientes:

- Gradiente Conjugado Escalado (**TRAINSCG**),
- BFGS Quasi Newton (**TRAINBFG**),
- Regularización Bayesiana (**TRAINBR**),
- Levenberg-Marquardt (**TRAINLM**).

El tipo de aprendizaje de la red neuronal es supervisado, lo cual conlleva a que se establezca previamente que valores son los deseados para cada conjunto de datos proporcionados en la capa de entrada.

5. Resultados y discusión

El entrenamiento de la red neuronal artificial propuesta en este estudio, se establece sobre la herramienta Matlab. Se crearon dos redes neuronales, las cuales fueron entrenadas por cada uno de los algoritmos de entrenamiento propuestos en la sección anterior. La primera red fue entrenada con un total de 30 x 1050 datos, de los cuales se tomaron aleatoriamente el 75% para entrenamiento, 15% para validación y 15% para pruebas. La segunda red fue entrenada con 30 x 700 datos, de los cuales 60% fueron a entrenamiento, 20% a validación y 20% a pruebas, sobre esta última red se simuló la clasificación con una matriz de pruebas conformada por 30 x 350, datos que no fueron incluidos en el entrenamiento a diferencia de la primera red.

En la Tabla 1 se presenta el rendimiento de la red neuronal (BPNN) entrenada con cada uno de los algoritmos y validada con el 15% de los datos aleatorios.

Tabla 1. Rendimiento de Algoritmos de entrenamiento medido en MSE. Conjunto de datos 30x1050

Algoritmo /Red	**Performance (MSE)**
TRAINBFG	0.0915
TRAINBR	0.0196
TRAINLM	0.0590
TRAINSCG	0.0675

En la Tabla 2 se presenta el rendimiento de la red neuronal (BPNN) entrenada con cada uno de los algoritmos, validada con el 20% de los datos aleatorios, y validada con un conjunto de datos diferentes a los de entrenamiento.

Tabla 2. Rendimiento de Algoritmos de entrenamiento medido en MSE. Conjunto de datos 30x700

Algoritmo/Red	Performance Train (MSE)	Performance Test (MSE)
TRAINBFG	0.0993	0.1276
TRAINBR	0.0233	0.1808
TRAINLM	0.0598	0.1615
TRAINSCG	0.0820	0.1221

Como se aprecia en la Tabla 1 y 2, el algoritmo TRAINBR proporcionó el mejor rendimiento en cuanto a entrenamiento de la red neuronal, pero el mejor rendimiento en las pruebas, lo proporcionó el algoritmo TRAINSCG. Esencialmente esto se debe a que los algoritmos TRAINBFG, TRAINSCG y TRAINLM realizaron un proceso de validación para que finalice el proceso de entrenamiento cuando encuentre 6 desaciertos, lo que hace que la red no se sobre-entrene, mientras que el algoritmo TRAINBR realiza un entrenamiento hasta cumplir la totalidad de las iteraciones (1000) o hasta que el valor del gradiente tienda a 0.

En la Tabla 3 se presenta la duración de cada algoritmo en el proceso de entrenamiento con datos de prueba independiente incluidos y sin datos de prueba independiente incluidos.

Tabla 3. Duración de entrenamiento de la Red Neuronal

Algoritmo/Red	Duración sin prueba incluida (Segundos)	Duración con prueba incluida (Segundos)
TRAINBFG	141	93
TRAINBR	755	569
TRAINLM	58	37
TRAINSCG	1	0

En la Tabla 3 se puede ver que el algoritmo TRAINSCG convergió bastante rápido debido a las validaciones que lleva a cabo, lo que frena el proceso de entrenamiento a tan solo un segundo. El algoritmo TRAINBR tardó bastante tiempo en el proceso de entrenamiento debido a que ejecuta todo el número de iteraciones (para este caso 1000).

En la Tabla 4 se presenta el número de iteraciones que fueron necesarias para que la red neuronal convergiera y asegurara buenos resultados de entrenamiento y pruebas.

Tabla 4. Numero de iteraciones realizadas por cada algoritmo

Algoritmo/Red	Iteraciones sin prueba (Nº/Total)	Iteraciones con prueba (Nº/Total)
TRAINBFG	52/1000	34/1000
TRAINBR	1000/1000	1000/1000
TRAINLM	17/1000	16/1000
TRAINSCG	71/1000	43/1000

El menor número de iteraciones lo realizó el algoritmo TRAINLM, valor que representa la efectividad de la red en converger rápido evitando exceso de procesamiento de la máquina.

En la Tabla 5 se aprecia el porcentaje de asertividad de cada algoritmo de entrenamiento con datos de pruebas aleatorios.

Tabla 5. Efectividad de entrenamiento y prueba de los algoritmos de entrenamiento. Conjunto de datos 30x1050

Algoritmo/Red	Entrenamiento (%)	Test (%)
TRAINBFG	69.3	53.2
TRAINBR	99.3	66.5
TRAINLM	87.2	62.7
TRAINSCG	80.5	67.7

En la Tabla 6 se muestra el porcentaje de asertividad de cada algoritmo de entrenamiento con datos de pruebas independientes.

Tabla 6. Efectividad de entrenamiento y prueba de los algoritmos de entrenamiento. Conjunto de datos 30x700

Algoritmo/Red	Entrenamiento (%)	Test(%) incluido	Test(%) independiente
TRAINBFG	64.3	52.9	51.1
TRAINBR	99.8	65.0	49.1
TRAINLM	95.2	56.4	47.4
TRAINSCG	71.7	64.3	54.0

En la Tabla 5 y 6, se aprecia que el algoritmo TRAINBR, proporcionó un alto porcentaje de acierto en el entrenamiento y en pruebas con datos incluidos y elegidos aleatoriamente, pero en la clasificación de pruebas independientes al conjunto de datos de entrenamiento, es el algoritmo TRAINSCG el que proporcionó mejores resultados.

Tomando en cuenta cada una de las características evaluadas en cada algoritmo de entrenamiento y sabiendo que las redes neuronales artificiales en el momento en que empiezan a aprender dejan de converger, se atribuye los mejores resultados al algoritmo de entrenamiento TRAINSCG, por su rapidez y buenos resultados de pruebas tanto incluidas en el conjunto de datos, como externas a estos.

En la Figura 4 se visualiza la matriz de confusión para la red entrenada con algoritmo TRAINSCG con datos de prueba incluidos y seleccionados aleatoriamente.

La Figura 4 expresa la diferencia entre el objetivo de la clasificación y los valores obtenidos del entrenamiento de la red neuronal, la cual fue capaz de clasificar un total de 67.7% de casos correctamente y obtuvo un promedio general de 76.3%. Los casos clasificados incorrectamente en realidad expresan la semana de maduración potencial de cada manzana en relación al cambio de sus características evaluadas, esto esencialmente porque cada fruta presenta un proceso de maduración diferente entre sí, lo que lleva a pensar que mientras una manzana debería ser clasificada en la cuarta semana de maduración, puede que sus características en realidad le atribuyan una clasificación en la primera semana en donde aún no hay cambio significativo de color, ni pérdida de peso, altura o diámetro.

Fig. 4. Matriz de confusión algoritmo TRAINSCG

6. Conclusiones y trabajo futuro

En este trabajo se propuso la creación de una red neuronal backpropagation multicapa con aprendizaje supervisado, entrenada con cuatro diferentes algoritmos de backpropagation (TRAINBFG, TRAINBR, TRAINLM y TRAINSCG), bajo dos esquemas diferentes, un conjunto de datos que incluyó un total de 30x1050 ejemplos de los cuales se extrajeron un 75% para entrenamiento, 15% para pruebas y 15% para validación, y otro con 30x700 datos de entrenamiento y 30x350 datos para pruebas independientes.

Los datos adquiridos para entrenamiento de la red provinieron de un previo tratamiento de las imágenes de 30 manzanas Royal Gala, equivalentes a 4200 imágenes tomadas durante 35 días y posteriormente transformadas a los modelos de color RGB, HSV y Lab.

Se analizaron los resultados obtenidos por el proceso de entrenamiento de cada algoritmo de aprendizaje tales como los valores de MSE, tiempo de ejecución, número de iteraciones, exactitud en el entrenamiento y en las pruebas. Los mejores resultados para la aplicación de esta red neuronal se obtuvieron con el algoritmo de Gradiente Conjugado Escalado (TRAINSCG)

Investigación posterior puede incluir la utilización de otras arquitecturas de red neuronal y otros algoritmos de entrenamiento. Complementar la clasificación teniendo de base características como los colores principales, las enfermedades de la fruta, y la calidad. Mejorar el rendimiento de los algoritmos de entrenamiento y extender su aplicación a otras frutas y/o áreas similares.

Referencias

1. Buitrago, M.A.: Manejo de Manzanas en Cosecha y Poscosecha. Bioplasma. Vol. Documentos, Núm. IV, p. 12 (1991)
2. Brezmes Llecha, J.J.: Diseño de una nariz electrónica para la determinación no destructiva del grado de maduración de la fruta. Universitat Politécnica de Catalunya (2002)
3. Chen, G., Dong, X.: From chaos to order: perspectives, methodologies and applications. Singapore. World Scientific (1998)
4. Fausett, L.V.: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice-Hall (1994)
5. Lin, C.T., Lee, C.S.: Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. Prentice-Hall, Inc. (1996)
6. Ben-Hanan, U., Peleg, K., Gutman, P.O.: Classification of fruits by a Boltzmann perceptron neural network. Automatica, Vol. 28, No. 5, pp. 961–968 (1992)
7. Nakano, K.: Application of neural networks to the color grading of apples. Computers and Electronics in Agriculture, Vol. 18, No. 2-3, pp. 105–116 (1997)
8. Zhang, Y., Wang, S., Ji, G., Phillips, P.: Fruit classification using computer vision and feedforward neural network. Journal of Food Engineering, 143, pp. 167–177 (2014)

Comparación de dos técnicas propuestas HS-CbCr y HS-ab para el modelado de color de piel en imágenes

Diana Alejandra Contreras Alejo, Francisco Javier Gallegos Funes

Instituto Politécnico Nacional, ESIME Zacatenco,
S.E.P.I. Doctorado en Comunicaciones y Electrónica, Cd. de México, México

dianalecontreras@gmail.com, fgallegosf@ipn.mx

Resumen. El contenido de este trabajo es la propuesta de dos técnicas para el modelado de color de piel en imágenes usando una combinación de espacios de colores HSV con YCbCr y HSV con CIELab. Se determina los intervalos de pixeles color no piel usando los componentes H y S, y los intervalos de pixeles color piel de los componentes Cb, Cr, a y b. Se compara los resultados de la técnica HS-CbCr que es la combinación de los componentes H y S (pixeles color no piel) con los componentes Cb y Cr (pixeles color piel), contra la técnica HS-ab que de la misma forma es la combinación de H y S pero con los componentes a y b (pixeles color piel). Se analiza la eficiencia de cada técnica para la segmentación de piel en base al porcentaje de detección de piel y no piel. Los resultados experimentales mostraron que la técnica HS-ab es mejor que la técnica HS-CbCr por la precisión además se descarta los obstáculos de luminancia pues en ambas técnicas se elimina la componente de luminancia.

Palabras clave: CIELab, crominancia, espacio de color, HSV, piel, YCbCr.

Comparison of Two Proposed Techniques HS-CbCr and HS-ab for the Modeling of Skin Color in Images

Abstract. The content of this paper is the proposal of two techniques for the modeling of skin color in images using a combination of spaces of colors HSV with YCbCr and HSV with CIELab. To determine the intervals of pixels non-skin color uses the components H and S, on the other hand, the intervals of pixels skin color uses the components Cb, Cr, a and b. There are compared the results of the technique HS-CbCr which is the combination of the components H and S (pixels non-skin color) with the components Cb and Cr (pixels skin color), against the technique HS-ab that of the same form is the combination of H and S but with the components a and b (pixels skin color). It analyzes the efficiency of each technique for the segmentation of skin based on the percentage of detection of skin and non-skin. The experimental results showed that technique HS-ab is better than the technique HS-CbCr because of the precision in addition the obstacles of luminance are discarded because in both techniques the luminance component is eliminated.

Keywords: CIELab, chrominance, color space, HSV, skin, YCbCr

1. Introducción

La detección de color de la piel humana en imágenes es una rama de investigación amplia en las áreas de visión por computador y procesamiento de imágenes. Algunas aplicaciones es el reconocimiento facial y/o corporal, vigilancia por video, interacción persona-ordenador (Human Computer Interaction-HCI), entre otras. Los problemas que se enfrentan para la detección de piel son el color, la orientación, la postura, la escala de la piel así como las condiciones de iluminación y los fondos complejos. La información de color es muy útil para extraer las regiones de piel en una imagen, además permite un procesamiento rápido y aporta robustez en la aplicación. De acuerdo a la literatura [1,2], los puntos esenciales para la detección de piel son la selección del espacio de color, el modelado de la distribución de color y depurar las regiones de la piel extraídas de la segmentación anterior.

En el primer punto, es de suma importancia la selección de espacio de color ya que determina la eficiencia del método para detectar el color de piel. Una desventaja que se presenta en el color es la sensibilidad al cambio de color de iluminación sobre todo en el espacio de color RGB. Una forma de resolver este problema es transformar la imagen RGB a otro espacio de color. Existe una variedad de espacio de colores para la clasificación de colores, en este caso nuestro interés es la segmentación de color de piel. La segmentación de color de piel determina si el pixel de color de una imagen es un color de piel o no es color de piel. Una buena segmentación de color de piel es aquella que segmenta cada color de piel ya sea negruzco, amarillento, pardo, blanquecino y aporta buenos resultados bajo las diferentes condiciones de luz. Los espacios de colores para el modelado de la piel se realizan mediante los componentes de crominancia porque se espera que la segmentación de color de piel pueda llegar a ser más resistente a las variaciones de iluminación si se descarta los componentes de luminancia [3]. En este documento los espacios de color HSV, YCrCb y CIELab se utilizan para la segmentación de color de la piel debido a que separan los componentes de crominancia y luminancia logrando una caracterización de los diferentes colores de piel [4,5].

En el segundo punto, el modelado del color de la piel es construir una regla de decisión que discrimine o diferencie entre los píxeles de una imagen que corresponden al color de la piel y aquéllos que no. Esto puede llevarse a cabo por definiciones explícitas de regiones de color, modelado no paramétrico de la distribución de piel o modelado paramétrico de la distribución de piel [6].

En el tercer punto, después de la segmentación de color de piel en la imagen puede contener ruido y algunas imperfecciones para mejorar la imagen y eliminar el ruido se hace uso de las operaciones morfológicas [7].

En este artículo abarca los dos primeros puntos, en el primer punto como elemento primordial se elige el espacio de color en este caso se optó por tres espacios de colores HSV, YCrCb y CIELab; posteriormente para el modelado de color de piel se analizaron los histogramas de las imágenes para identificar los pixeles de color de piel y por último se realiza el algoritmo para detectar las regiones de piel en las imágenes. La aportación del artículo es la presentación de dos técnicas de modelos de color para la detección de color de piel. Ambas técnicas utilizan el espacio de color HSV que proporciona información adicional de tono y crominancia de una imagen con la finalidad de mejorar la discriminación entre los píxeles de la piel y píxeles no de piel [8]. En este trabajo el espacio de color HSV se usó para detectar los pixeles de color no piel posteriormente,

en un método se aplica el espacio de color YCbCr para detectar los pixeles de color de piel y en el otro método se usa el espacio de color CIELab. Finalmente se realiza un análisis de histogramas de imágenes de piel de la base de datos SFA [9] para determinar el número de umbral de cada espacio de color que se implementa en el algoritmo.

2. Espacio de color para el modelado de color de piel

Se define espacio de color como una representación matemática de un conjunto de colores [10]. Hay una variedad de espacios de colores disponibles se dividen en cuatro grupos, algunos espacios de colores más usados se muestran en la Tabla 1.

Tabla 1. Grupos de espacios de colores más comunes

Grupo	Espacio de colores
Modelo de color básico	RGB, RGB Normalizado, XYZ
Modelo de color perceptual	HIS, HSV, HSL, TSL, TSV
Modelo de color ortogonal	YCbCr, YIQ, YES , YUV
Colores perceptualmente uniforme	CIELab, CIELuv,CIE XYZ, CIE-xy Y

La mayor parte de la investigación en la detección de color de piel se basa en los espacios de color HIS, HSV, YCbCr y CIELab [3, 11, 12]. A continuación se explicará brevemente los espacios de color HSV, YCbCr y CIELab.

2.1. HSV

El problema del espacio de color RGB (en inglés Red, Green and Blue) no proporciona la información correcta sobre el color de piel debido a los efectos de luminancia. HSV ofrece información de tono (en inglés Hue) que define el color dominante (como el rojo, verde, morado y amarillo) de una zona y tiene un rango de 0° a 360°, saturación (en inglés Saturation) que mide el colorido de un área en proporción a su brillo y tiene un rango de 0% a 100%, y valor (en inglés Value) se refiere al brillo del color y provee una noción acromática de la intensidad de color. Los componentes H y S entregan información útil para la discriminación en la búsqueda de piel, es por eso que se emplea para detectar a los pixeles de color no piel [13]. Para convertir el espacio de color RGB al HSV se usan las siguientes expresiones matemáticas:

$$H = \arccos \frac{1/2\left[(R-G)+(R-B)\right]}{\sqrt{\left[(R-G)^2+(R-B)(G-B)\right]}}, \tag{1}$$

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B}, \tag{2}$$

$$V = \frac{1}{3}(R+G+B), \tag{3}$$

donde H es tono, S es saturación, V es valor, R es el componente rojo, G es el componente verde y B es el componente azul. En la Fig. 1, se muestra el modelo HSV

donde su sistema coordenado es cilíndrico y su subespacio es una pirámide de base hexagonal.

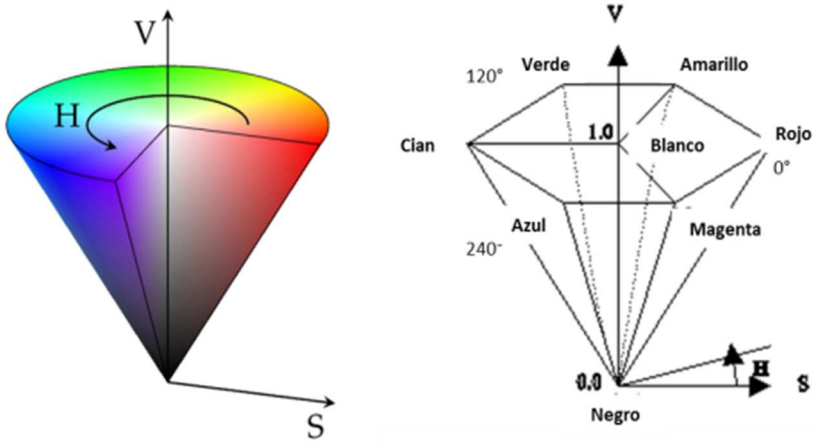


Fig. 1. Modelo HSV. La tonalidad se representa por valores del grado de un ángulo (0°-360°), la saturación se representa como la distancia del eje del brillo (negro-blanco) y el valor representa la altura en el eje (negro-blanco)

2.2. YCbCr

En este espacio de color la información de la luminancia está representada por una solo componente Y, por otro lado la información de color es almacenada en los componentes Cb y Cr. Cb es la componente de crominancia azul y se define como la diferencia entre la componente azul y el valor de referencia ver ecuación (5). Cr es la componente de crominancia roja y como se ve en la ecuación (6) es la diferencia entre la componente roja y el valor de referencia [14]. Para transformar el espacio de color RGB a YCbCr se lleva acabo con las siguientes ecuaciones:

$$Y = 0.299R + 0.587G + 0.114B, \tag{4}$$

$$Cb = B - Y, \tag{5}$$

$$Cr = R - Y, \tag{6}$$

donde R es el componente rojo, G es el componente verde y B es el componente azul. La Fig. 2 expone una representación gráfica del espacio de color YCbCr.

2.3. CIELab

Es un modelo cromático para describir todos los colores que puede percibir el ojo humano. C.I.E. (Commission Internationale d'Eclairage) es un organismo encargado de especificar los estándares de color como CIEXYZ, CIELuv, y CIELab. La componente L es la luminancia (va de negro a blanco) y los componentes a,b son

colores cromáticos (en a oscila entre rojo a verde mientras en b oscila entre azul a amarillo) [15]. El modelo CIELab se modela en la Fig. 3.

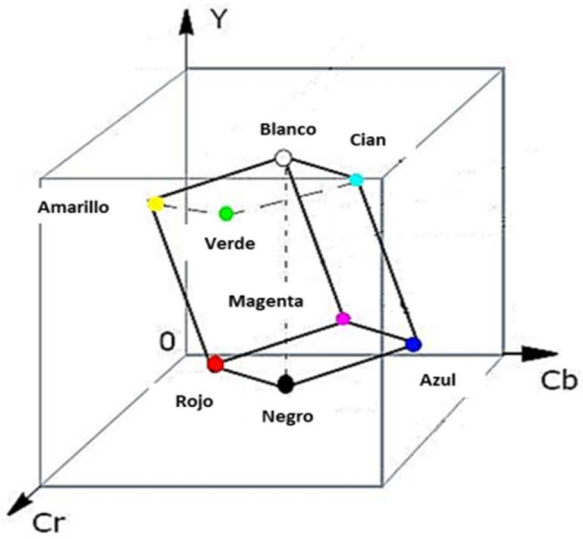


Fig. 2. Modelo YCbCr. Todos los colores posibles de RGB ocupan sólo una parte del espacio de color YCbCr (cubo interior). En el cubo grande representa todos los valores posibles de YCbCr

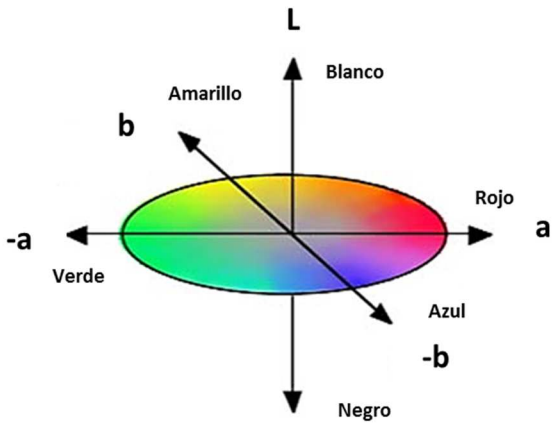


Fig. 3. Modelo CIELab. L = 0 proporciona el color negro y L = 100 proporciona el color blanco. Los valores -a<0 indican verde mientras que los valores a>0 indican rojo. Los valores de -b<0 indican azul y los valores de b> 0 indican amarillo

3. Metodología

El procedimiento para el desarrollo de este trabajo es el siguiente:

3.1. Cálculo de umbral

Se obtienen 40 imágenes de diferentes colores y texturas, y se descargan 40 imágenes de la base de datos SFA que contiene una variedad de colores de piel. En el primer caso, cada imagen se transforma del modelo de color RGB a HSV obteniendo los histogramas de los componentes H y S. En dichos histogramas se tienen los valores máximos y mínimos de los distintos tonos de piel de cada componente. Con todos los valores que se obtuvieron de las 40 imágenes se tiene el valor promedio para finalmente tener el intervalo del componente H y S. Para el segundo caso es el mismo procedimiento excepto que las imágenes se transforman al modelo de color YCbCr y CIELab; por lo tanto se tienen los intervalos de los componentes Cb, Cr, a y b. El procedimiento se simplifica en la Fig. 4.

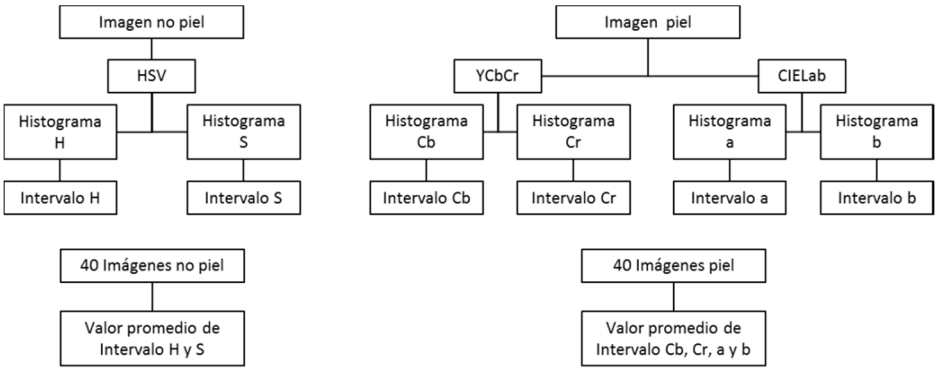


Fig. 4. Diagrama para cálculo de umbral. En la parte de arriba se describe el proceso de una sola imagen tanto de la no piel como la de piel. En la parte de abajo indica que después de obtener los intervalos de las 40 imágenes se obtiene el intervalo promedio

3.2. Técnicas HS-CbCr y HS-ab

Después de conocer el umbral de los intervalos de cada espacio de color. Se desarrolla un algoritmo que nos va a permitir detectar las regiones de piel de una imagen. Este algoritmo se implementa en MATLAB (R2015b-The MathWorks). Este algoritmo consiste en cargar la imagen al programa, transformar la imagen en espacio de color HSV e identificar los pixeles no piel con los intervalos H y S para descartarlos; después con esa misma imagen se convierte al espacio de color YCbCr para seleccionar los pixeles de color piel con el intervalo Cb y Cr y se despliega la imagen con las regiones de solo piel, esta es la técnica HS-CbCr. Por otro lado la técnica HS-ab tiene el mismo procedimiento descrito anteriormente a excepción que para identificar los pixeles de color de piel es con el intervalo a y b. Por último en ambas técnicas, se calcula el porcentaje de piel de la imagen contando todos los pixeles que sean diferentes de cero o diferentes de negro, es decir, la zona más grande de piel en la imagen (A) entre el número total de pixeles de color de piel (B). Esto se define en la ecuación (7).

$$\%\ piel = \frac{A}{B} * 100\%. \tag{7}$$

En la Fig. 5, se presenta el diagrama de flujo del algoritmo descrito previamente.

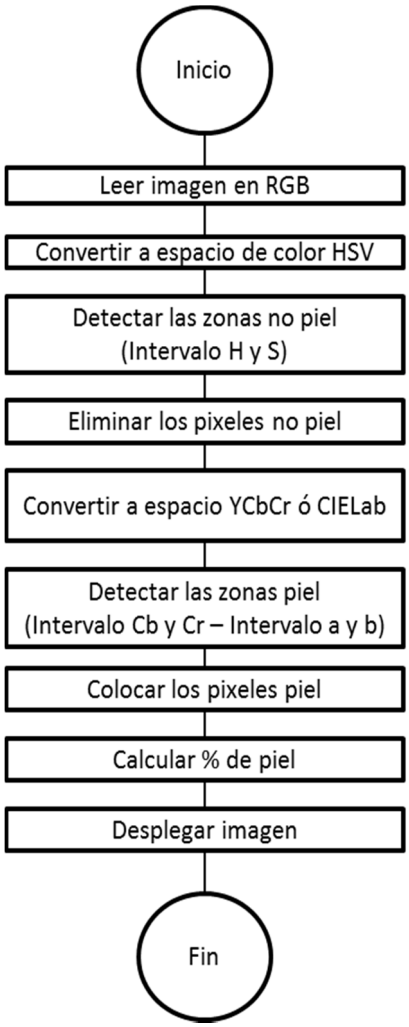


Fig. 5. Algoritmo de las técnicas HS-CbCr y HS-ab. Como se observa es el mismo algoritmo para las dos técnicas sólo cambia la conversión de modelo de color YCbCr y CIELab con sus respectivos intervalos

3.3. Evaluación

Después La evaluación se realiza con el fin de determinar qué componentes de crominancia producen los mejores resultados de la detección de piel. Esta evaluación consiste en obtener el número de pixeles que son clasificados de color de piel tanto para el espacio de color YCbCr y CIELab de las 40 imágenes de piel, también se obtiene el número total de pixeles de las imágenes. Esta evaluación está expresada en la ecuación (8):

$$\%C = \frac{No.pixelesdecolorpiel}{No.pixelestotal} * 100\%. \quad (8)$$

4. Resultados

Las 40 imágenes color no piel y 40 imágenes color piel que se usaron se presenta en la Fig. 6. Cada imagen se transformó al espacio de color HSV y se obtuvo su histograma de las canales H y S para obtener los valores mínimos y máximos de cada canal. Este proceso se repite para el espacio de color YCbCr y para el espacio de color CIELab, un ejemplo se observa en la Fig. 7 donde se convierte una imagen de color piel en el espacio de color YCbCr presentando sus respectivos histogramas de cada canal. En este caso el intervalo es Cb=[110,114] y Cr=[138,142]. Al tener todos los valores mínimos y máximos de cada espacio de color, se tiene el valor promedio para tener el intervalo de las imágenes no piel en modelo HSV: 0 < H < 0.2 y 0.15 < S < 0.9, el intervalo de las imágenes piel en modelo YCbCr: 88 < Cb < 130 y 127 < Cr < 175, y el intervalo de las imágenes piel en modelo CIELab: 142 < a < 225 y 115 < b < 177.

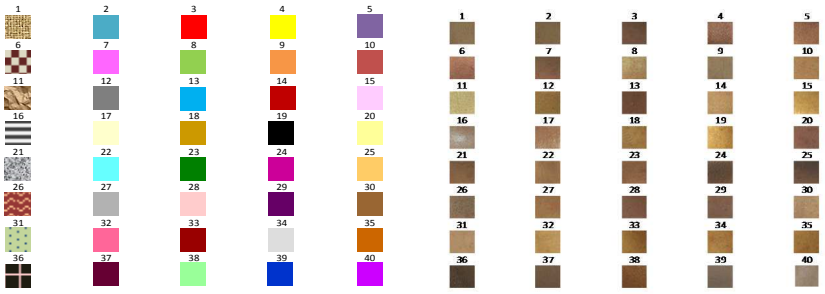


Fig. 6. Imágenes. Las imágenes color no piel son aquellas imágenes con una variedad de colores y texturas mientras las imágenes color piel son imágenes reales de diversos tipos de piel

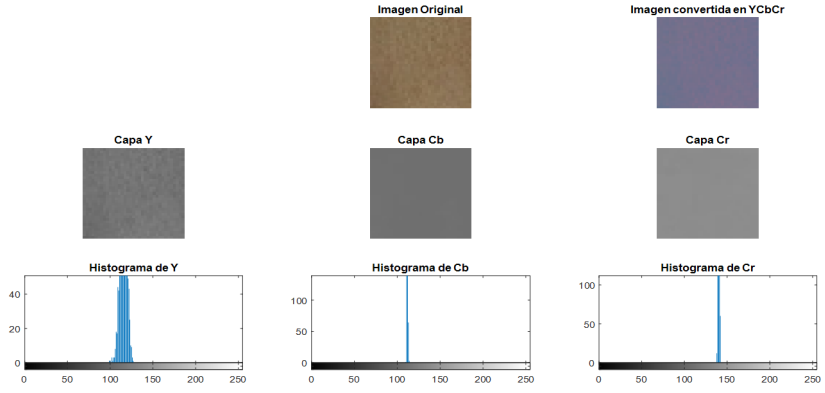


Fig. 7. Histogramas de las capas de color YCbCr en una imagen de piel. La imagen original en RGB se convierte en espacio de color YCbCr, y se tiene el histograma de Y, Cb y Cr; en este caso los valores que nos interesan son los de componentes de crominancia es decir Cb y Cr

Los intervalos se implementaron en el algoritmo, se usaron 20 imágenes reales de personas para la detección de piel. En la Tabla 2, se expone los resultados de los porcentajes de piel de cada técnica. Y se observa una gran diferencia en los porcentajes de detección de piel de la técnica HS-CbCr con respecto a la técnica HS-ab, entonces se podría decir que al comparar ambas técnicas, la primera detecta mayor cantidad de regiones de piel en la imagen.

Tabla 2. Porcentaje de detección de piel y no piel de la técnica HS-CbCr y de la técnica HS-ab

No.	Técnica HS-CbCr % de piel	Técnica HS-ab % de piel	% de no piel
1	28.31	11.87	29.25
2	52.88	4.19	52.89
3	27.42	4.29	27.69
4	34.44	16.68	36.22
5	59.88	21.67	59.99
6	53.57	10.80	53.67
7	51.31	8.56	51.57
8	8.49	0.54	8.77
9	39.79	4.60	41.07
10	9.26	4.31	9.27
11	23.14	5.03	25.61
12	32.94	5.70	34.19
13	21.42	5.42	21.55
14	74.97	10.14	75.35
15	44.13	35.17	44.44
16	69.31	6.24	70.19
17	13.07	1.62	13.30
18	63.79	30.03	63.96
19	39.14	28.86	39.90
20	31.97	4.20	32.30

Sin embargo en la Fig. 8, se muestran cinco imágenes ya procesadas con ambas técnicas; y a simple vista la mejor detección de piel es en la técnica HS-ab ya que detecta sola la piel mientras en la técnica HS-CbCr presenta falsas alarmas pues detecta la piel y otros factores (fondo, cabello, etc.) que presentan colores similares a la piel.

Tabla 3. Porcentaje de detección de piel en los componentes de crominancia

Espacio de color	Crominancia	%C
YCbCr	CbCr	33.51
CIELab	ab	34.80

Para evaluar cuantitativamente que espacio de color detecta más preciso el color de piel se aplica la ecuación (8). En la Tabla 3 se presenta los porcentajes de detección de piel en los componentes de crominancia; y se concluye que el espacio de color CIELab es más preciso en la detección de piel con respecto a YCbCr.

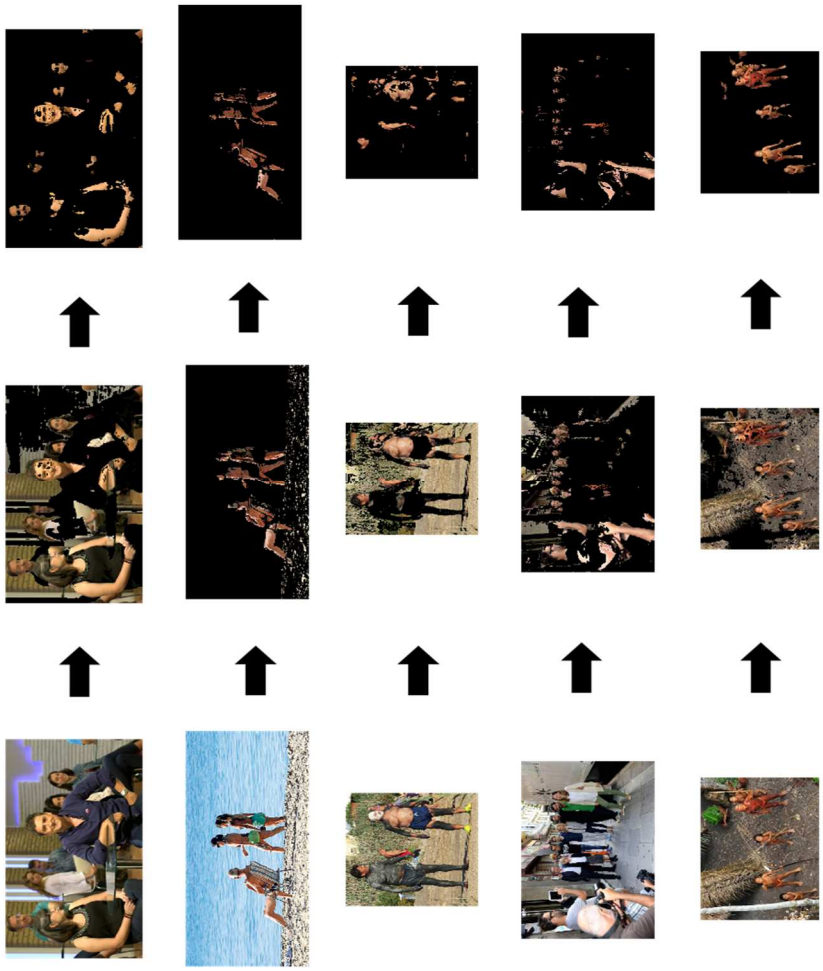


Fig. 8. Imágenes con aplicación de las técnicas propuestas. De izquierda a derecha, en la primera columna se muestran las imágenes reales (originales, en la segunda columna son las imágenes con la aplicación de la técnica HS-CbCr, y en la última las imágenes son procesadas con la técnica HS-ab

5. Conclusiones

Hoy en día, el avance de investigación en los métodos de detección de piel es necesario en múltiples disciplinas. Una técnica apropiada para detectar los pixeles de

color de piel y no color de piel es vital para imágenes con piel de diferente tipos de piel como el rosa, amarillo, blanco, marrón oscuro y marrón claro. En este trabajo, se presenta la innovación y comparación de dos técnicas para el modelado de color de piel. La segmentación de regiones de piel fue desarrollada usando los intervalos H y S de no piel del modelo de color HSV con el espacio de color YCbCr y CIELab del cual en ambas técnicas demuestra una clara discriminación entre regiones de piel y no piel. Los resultados experimentales mostraron que nuestro nuevo enfoque de modelado de color de piel fue capaz de lograr la detección de piel con éxito en todas las imágenes. De acuerdo con los resultados, en la primera técnica (HS-CbCr) se presenta un mayor porcentaje de detección de piel comparado con la segunda técnica (HS-ab) pero eso no indica que es mejor la primera técnica; ya que en todos de los casos se presentó una diferencia mínima respecto al porcentaje de pixeles no piel con el porcentaje de detección de piel de HS-CbCr; eso significa que la técnica HS-CbCr presenta una confusión de identificar el color por ejemplo detecta los colores del cabello castaño, güero como color de piel. Sin embargo en la técnica HS-ab su porcentaje de detección de color de piel es mucho menor que el porcentaje de pixeles no piel además se observa que descarta los colores de cabello u otros parecidos al de la piel y el resultado de %C del espacio de color CIELab fue mayor que YCbCr eso indica que los componentes de crominancia en el primer espacio de color son más precisos en la detección de color de piel. Por lo tanto se concluye que es fiable la segunda técnica HS-ab para la detección de color de piel además se reafirma lo mencionado en la literatura [3] con respecto a que la segmentación del espacio de color CIELab es mejor que YCbCr que da más información de color que otros espacio de colores por lo que es más preciso.

Agradecimientos. Los autores agradecen al Instituto Politécnico Nacional de México (IPN) y al Consejo Nacional de Ciencia y Tecnología de México (CONACYT) por la ayuda y el apoyo para desarrollar la Investigación. Como la colaboración y apoyo otorgado por parte de los Doctores del posgrado de la S.E.P.I. de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Profesional Zacatenco.

Referencias

1. Vezhnevets, V., Sazonov V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: GRAPHICON'03, pp. 85–92 (2003)
2. Prema, C., Manimegalai, D.: Survey on Skin Tone Detection using Color Spaces. International Journal of Applied Information Systems, Published by Foundation of Computer Science, New York, USA, Vol. 2, No. 2, pp. 18–26, (2012)
3. Amanpreet, K., Kranthi, B.: Comparison between YCbCr Color Space and CIELab Color Space for Skin Color Segmentation. International Journal of Applied Information Systems Published by Foundation of Computer Science, New York, USA, Vol. 3, No. 4, pp. 30–33, July (2012)
4. Phung, S.L., Bouzerdoum, A., Chai, D.: Skin segmentation using color pixel classification: analysis and comparison. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 1 (2005)
5. Sabottka, K., Pitas, I.: Segmentation and Tracking of Faces in Color Images. In: AFGR'96, Killington, Vermont, pp. 236–241 (1996)

6. Aznaveh, M.M., Mirzaei, H., Roshan, E., Saraee, M.: A new color based method for skin detection using RGB vector space. In: Human System Interactions, Conference on, Krakow, pp. 932–935 (2008)
7. Bhata, V.S., Pujaria, J.D.: Face detection system using HSV color model and morphing operations. International Journal of Current Engineering and Technology, pp. 200–204 (2013)
8. Tabassum, M.R., Gias, A.U., Kamal, M.M., Muctadir, H.M., Ibrahim, M., Shakir, A.K., Imran, A., Islamm, S., Rabbani, G., Khaled, S.M., Islam, S., Begum, Z.: Comparative Study of Statistical Skin Detection Algorithms for Sub-Continental Human Images. Information Technology Journal, Vol. 9, No. 4, pp. 811–817 (2010)
9. Casati, J.P.B., Moraes, D.R., Rodrigues, E.L.L.: SFA: A Human Skin Image Database based on FERET and AR Facial Images. In: IX Workshop de Vision Computational (2013)
10. Fairchild, M.D.: Color Appearance Models. John Wiley & Sons. ed.3, p. 472 (2013)
11. Lee, Y.J., Yoo, S.I.: An elliptical boundary model for skin color detection. In: International Conference on Imaging Science, Systems, and Technology, pp. 24–27 (2002)
12. Chai, D., Bouzerdoum, A.: A Bayesian approach to skin color classification in YCbCr color space. In: Institute of Electrical and Electronics Engineers IEEE Region Ten Conference (TENCON), Vol. 2, pp. 421–424 (2000)
13. Qiong, L., Guang-zheng, P.: A robust skin color based face detection algorithm. In: Informatics in Control, Automation and Robotics (CAR), 2nd International Asia Conference on, Wuhan, pp. 525–528 (2010)
14. Phung, S.L., Bouzerdoum, A., Chai, D.: A novel skin color model in YCbCr color space and its application to human face detection. In: Image Processing Proceedings International Conference on, Vol. 1, pp. I-289-I-292 (2002)
15. Sharma, G., Bala, R.: Digital Color Imaging Handbook: Electrical Engineering & Applied Signal Processing Serie. CRC Press, pp. 32–45 (2002)

Uso de redes neuronales pulsantes para mejorar el filtrado de imágenes contaminadas con ruido Gaussiano

Estela Ortiz Rangel, Manuel Mejía-Lavalle, Humberto Sossa Azuela

Centro Nacional de Investigación y Desarrollo Tecnológico,
Departamento de Ciencias Computacionales, Cuernavaca, Morelos,
México

Instituto Politécnico Nacional, CIC, Ciudad de México
México

{estela_or, mlavalle} @cenidet.edu.mx, hsossa@cic.ipn.mx

Resumen. Se propone un algoritmo llamado ICM-TM para reducir el efecto de ruido gaussiano en imágenes en escala de grises basado en una Red Neuronal Artificial tipo Pulso-Acoplada simplificada llamada Intersection Cortical Model (ICM). Una matriz de tiempos (TM) concentra la información respectiva al número de iteración donde se activa por primera vez la neurona correspondiente a cada pixel; basándose en los tiempos de activación de las neuronas se establece un criterio de filtrado selectivo combinando los operadores mediana y promedio. El desempeño del algoritmo propuesto fue evaluado experimentalmente, con distintos grados de ruido gaussiano y los resultados de las simulaciones muestran que la efectividad del método es superior al filtro de mediana convencional, al filtro Wiener y a la técnica Pulse-Coupled Neural Networks with the Null Interconnections (PCNNNI); los resultados están representados por el parámetro Peak Signal to Noise Ratio (PSNR) principalmente.

Palabras clave: Intersection Cortical Model (ICM), ruido gaussiano, filtro Wiener, Peak Signal to Noise Ratio (PSNR).

Using Pulsed Neural Networks to Improve Filtering of Images Contaminated with Gaussian Noise

Abstract. An algorithm called ICM-TM is proposed to reduce the effect of Gaussian noise in grayscale images based on a kind of Artificial Neural Networks type Pulse-Coupled simplified called Intersection Cortical Model (ICM). A Time Matrix (TM) provides the respective information to the iteration number where first activated neuron corresponding to each pixel; a selective filtering criteria is established combining the median and average operators based on the neurons activation times. The performance of the proposed algorithm was evaluated experimentally with varying degrees of Gaussian noise and the results of the simulations show that the effectiveness of the method is superior to the median filter, Wiener filter and to the Pulse-Coupled Neural Networks with the Null

Interconnections (PCNNNI); the results are mainly represented by the parameter Peak Signal to Noise Ratio (PSNR).

Keywords: Intersection cortical model (ICM), Gaussian noise, Wiener filter, peak signal to noise ratio (PSNR).

1. Introducción

El ruido en las imágenes digitales es información no deseada que las modifica; puede deberse a defectos en los dispositivos de captura, en los medios de transmisión o almacenamiento y causa problemas para el procesamiento posterior de las imágenes.

El ruido aditivo gaussiano q es un modelo que simula la afectación aleatoria de todos los pixeles de una imagen con valores uniformemente distribuidos, donde su función de densidad de probabilidad $p_q(x)$ está dada en términos del promedio μ y la varianza σ^2de una variable aleatoria x como se expresa en la ec. 1 [1]. Este tipo de ruido es muy común en las imágenes digitales y debido a sus características es difícil de eliminar por completo.

$$\mathrm{p_q(x)} = (2\pi\sigma^2)^{-1/2}\mathrm{e}^{-(\mathrm{x}-\mu)^2/2\sigma^2}. \tag{1}$$

Existen métodos para la reducción del ruido gaussiano que pueden ser clasificados como filtros espaciales y frecuenciales. Los filtros espaciales pueden ser lineales tales como el filtro promedio y el gaussiano y los no lineales como el filtro de mediana y el filtro sigma.

El filtro *Wiener* $H(u, v)$ como se expresa en la ec. 2 [2] es un filtro frecuencial que se basa en la reducción del error cuadrático medio y mejora sustancialmente la calidad de una imagen con ruido gaussiano, sin embargo requiere del cálculo del espectro de energía de la imagen ideal $S_w(u, v)$ y del ruido $S_f(u, v)$, de la estimación de una función de degradación $D(u, v)$ y su conjugado $D^*(u, v)$ para realizar el filtrado.

$$\mathrm{H(u,v)} = \frac{\mathrm{D^*(u,v)}}{\mathrm{D^*(u,v)D(u,v)} + \frac{\mathrm{S_w(u,v)}}{\mathrm{S_f(u,v)}}}. \tag{2}$$

Algunas modificaciones han sido propuestas a este filtro con el fin de mejorar su desempeño y generalizar su aplicación [3]. El filtro *Sigma* también es uno de los más efectivos en la eliminación del ruido gaussiano y es ampliamente utilizado por su simplicidad, no obstante su capacidad de preservar bordes aún se sigue mejorando, por ejemplo por medio de técnicas difusas [4].

También se ha propuesto técnicas que utilizan la información de los bordes de la imagen contaminada, donde se aplican los principios de similaridad, no obstante su proceso computacional es largo [5, 6] y por otro lado están las técnicas basadas en transformadas *wavelet* son promisorias pero conllevan largos procesos de cálculo, lo cual dificulta su implementación en tiempo real y en sistemas embebidos [7].

Una forma distinta de enfrentar el problema de eliminación del ruido gaussiano en imágenes digitales ha surgido de la exploración experimental, tal es el caso de las Redes

Neuronales Artificiales de tercera generación llamadas Redes Neuronales Pulso-Acopladas o por sus siglas en inglés PCNN (*Pulse Coupled Neural Networks*), las cuales han sido empleadas de modo eficiente para el procesamiento de imágenes en diversas tareas como la segmentación, la clasificación, la identificación de imágenes, entre otras [8].

Las redes tipo PCNN son un modelo matemático, propuesto por Eckhorn, basado en la frecuencia de activación de las neuronas de la corteza visual de los mamíferos [9]. El tiempo y frecuencia de activación de las neuronas han sido utilizados para procesar imágenes gracias al modelo computacional simplificado de PCNN propuesto por Ranganath y Kuntimad [10].

Las propiedades del modelo de PCNN permiten a cada neurona corresponder con un pixel y relacionar su nivel de gris con el de sus vecinos, de modo que al iterar la Red Neuronal el tiempo de activación de cada neurona se registra en una matriz. Esta información puede ser utilizada para la detección de los pixeles con ruido y la aplicación de una técnica selectiva de filtrado de ruido gaussiano [11, 12], e incluso para la reducción del ruido gaussiano y de sal y pimienta mezclados [13, 14].

Diversos modelos simplificados de PCNN han sido propuestos para trabajar computacionalmente con imágenes; dos de las principales variaciones son *Intersection Cortical Model* (ICM) y *Pulse-Coupled Neural Networks with the Null Interconnections (PCNNNI)* [10]. Ma [15] ha utilizado estos métodos combinándolos con el operador de mediana, promedio y morfológico o con el filtro *Wiener* y la matriz de tiempos para reducir el ruido de Sal y Pimienta y el ruido gaussiano en imágenes digitales.

El método ICM-TM que se propone consiste en utilizar una red ICM para generar la matriz de tiempos y aplicar selectivamente los operadores mediana y promedio para suprimir el ruido gaussiano en imágenes digitales en escala de grises, de forma que se logre una significativa reducción del ruido con un método computacional simple que puede ser implementado para aplicaciones con sistemas embebidos.

2. Modelo de PCNN e ICM

Las Redes Neuronales Pulso-Acopladas son un sistema que emula efectivamente a las neuronas biológicas de la corteza visual de los mamíferos y han sido aplicadas en variedad de dominios, especialmente en el procesamiento de imágenes han sido útiles para remoción de ruido, reconocimiento de objetos, optimización, adelgazamiento, segmentación, fusión, identificación y remoción de sombras [8,11].

En el ámbito del procesamiento de imágenes las diferencias entre las redes Neuronales Artificiales tradicionales y las Redes Neuronales de tercera generación son evidentes tanto en su configuración como en su operación.

Las redes tipo PCNN no requieren entrenamiento ya que su función es clasificar los pixeles por sus niveles de intensidad, en este modelo cada pixel de la imagen corresponde a una neurona y su valor es introducido a la red por medio de una señal llamada *feeding*. El umbral de activación de las neuronas es dinámico, además cada neurona recibe información de sus vecinas a través de una sinapsis, lo cual se conoce como *linking*. *Feeding* y *linking* se combinan para formar el potencial interno de la neurona, que al ser comparado con el umbral produce salidas binarias. Estas

características hacen que las neuronas correspondientes a pixeles vecinos con valores de intensidad similares se activen al mismo tiempo en ciertas regiones, a lo que se denomina activación de pulsos síncrona [16].

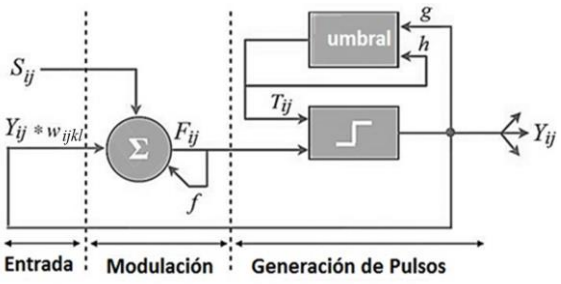


Fig. 1. Diagrama general de ICM [14]

El modelo original de PCNN tiene algunas limitantes en la práctica debido al gran número de interconexiones entre neuronas y al establecimiento de parámetros de operación; por este motivo fue obtenido un modelo iterativo simplificado de PCNN que permitiera extender el uso de estas redes en el procesamiento de imágenes.

ICM es un tipo de PCNN más simple (Fig. 1), donde no se considera *linking* en las neuronas y donde el *feeding* F_{ij} mantiene su salida con un factor de decaimiento dado por f. La señal F_{ij} está compuesta por la última salida del vecindario de neuronas Y_{ij} ponderada por una matriz w_{ijkl} (generalmente gaussiana) y por el estímulo externo de entrada S_{ij} (nivel de gris de cada pixel normalizado entre 0 y 1). El umbral dinámico para cada pixel T_{ij} crece obedeciendo a h cuando su salida se activa y mantiene su estado previo con una atenuación dada por g, lo cual da origen a la formación de pulsos.

El modelo computacional iterativo ICM puede ser descrito por medio de las siguientes funciones (3)-(5) [15]:

$$F_{ij}\,[n] = fF_{ij}\,[n-1] + \sum w_{ijkl}Y_{kl}[n-1] + S_{ij}, \tag{3}$$

$$T_{ij}[n] = gT_{ij}[n-1] + hY_{ij}[n-1], \tag{1}$$

$$Y_{ij}[n] = \begin{cases} 1 & \text{si } F_{ij}[n] > \mathrm{T}_{ij}[n] \\ 0 & en\ otro\ caso \end{cases}, \tag{2}$$

donde n es la iteración actual, w_{ijkl} es la matriz de pesos sinápticos que liga una neurona con sus vecinas y finalmente f, g y h son coeficientes de ajuste, donde típicamente $g < 1.0$, $f < g$ y h es un valor grande.

3. La matriz de tiempos (TM)

Las redes tipo PCNN pueden ser utilizadas para determinar la posición de los pixeles ruidosos con base en las neuronas activadas a la salida de la red por iteración y, al aplicar un operador de mediana que elimine los valores más altos y más bajos, es posible eliminar el ruido de Sal y Pimienta.

No obstante para el filtrado del ruido gaussiano es necesario emplear otra técnica ya que todos los pixeles han sido afectados en algún grado, por lo que se introduce la matriz de tiempos de las mismas dimensiones que la imagen a tratar; ella contiene información relacionada con la estructura espacial de la imagen con ruido, es decir, realiza un mapeo de la información espacial en una secuencia temporal [11].

La matriz de tiempos obtenida a partir de una red neuronal con interconexión nula PCNNNI [15] ha sido utilizada para detectar los pixeles ruidosos y procesada para reducir el ruido gaussiano combinándola con otros métodos [12]. El modelo PCNNNI no considera la señal de *linking* y elimina la influencia de la matriz de pesos sinápticos que liga a una neurona con sus vecinas, por lo que el potencial interno de una neurona, su umbral y salida únicamente dependen de la intensidad de su pixel correspondiente, además de que dicha técnica modifica el rango dinámico de la imagen para reducir el número de iteraciones necesarias y aplica cinco criterios de filtrado [15].

El método propuesto ICM-TM consiste en obtener la matriz de tiempos conservando la información de la relación espacial entre pixeles, de modo que las diferencias en la intensidad de los pixeles de la imagen originen diferencias en la secuencia de activación de sus respectivas neuronas. Para conservar el tiempo de activación de cada neurona se define una matriz M_{ij} que puede ser descrita como [11]:

$$M_{ij}[n] = \begin{cases} n & \text{si } Y_{ij}[n] = 1 \\ M_{ij}[n-1] & en\ otro\ caso \end{cases}. \tag{6}$$

El algoritmo ICM-TM conserva el rango dinámico de la imagen y complementa el proceso de filtrado obteniendo la información de la matriz de tiempos para cualquier número de iteraciones, suavizando regiones de 5x5 pixeles con el filtro promedio para suavizar la imagen o bien conservando información de la imagen pero suprimiendo valores extremos mediante el operador de mediana.

4. Descripción del algoritmo propuesto ICM-TM

En nuestro método todas las neuronas de ICM están ligadas mutuamente del mismo modo y sus salidas tienen dos estados posibles, activado y no activado. Este modelo es más rápido que PCNN debido a que implica menos ecuaciones y es más adaptable ya que tiene menos parámetros que ajustar.

El proceso principal del algoritmo ICM-TM (Fig. 2) es como sigue:

Paso 1. Pasar los valores de gris de los pixeles de la imagen con ruido O_{ij}, normalizados entre 0 y 1 a la red ICM por medio de S_{ij}.

Paso 2. Cuando una neurona en la posición (i,j) tome el valor de activada por primera vez $(Y_{ij}[n] = 1)$ registrar el número de iteración actual n en la matriz de tiempos $M_{ij}[n]$ en el lugar correspondiente al pixel procesado (i,j), omitiendo la primera iteración.

Paso 3. Continuar iterando la red hasta que todos los pixeles se hayan activado por lo menos una vez, es decir cuando todos los elementos en $M_{ij}[n]$ son distintos de cero.

Paso 4. Con base en la matriz de tiempos $M_{ij}[n]$ aplicar el siguiente criterio:

a) Si el tiempo de activación de una neurona es igual a la mediana de los tiempos en una vecindad de 25 neuronas centradas en ésta, el valor del pixel en la imagen filtrada será el valor truncado en entero resultante de aplicar el operador promedio sobre esta vecindad en la imagen con ruido O_{ij}.

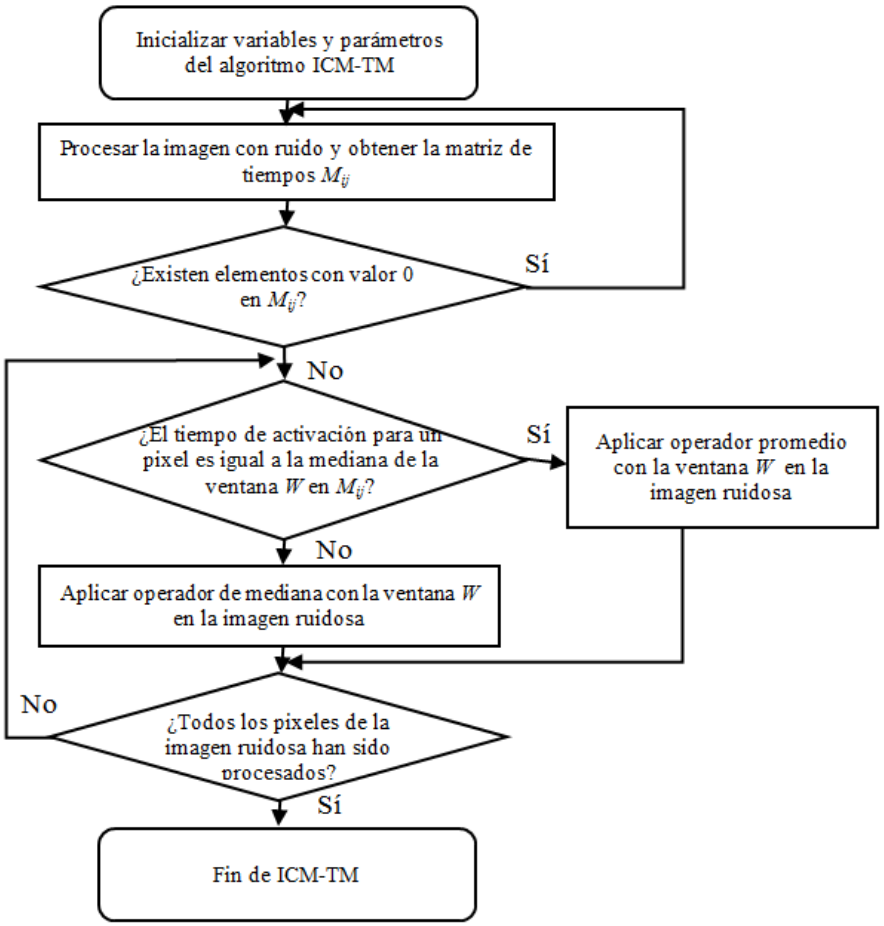


Fig. 2. Diagrama del algoritmo propuesto

b) Si no el valor del pixe**l** de la imagen filtrada será el resultante de aplicar el operador de mediana sobre una vecindad de 25 pixeles centrada en este pixel de la imagen con ruido O_{ij}. En este caso la ventana W es de 5x5, seleccionada para un mejor efecto de suavizado.

5. Experimentación y resultados

Las pruebas experimentales fueron realizadas con las imágenes en escala de grises de 255 valores: *Lena*, *Peppers* y *Baboon*; con tamaño de 512x512 pixeles.

Durante el proceso de experimentación, se introdujo a las imágenes ruido gaussiano de promedio 0 y varianza entre 0.01 y 0.09. La imagen O_{ij} normalizada entre 0 y 1 se ingresó al ICM como $S_{ij}[n]$. Las matrices $F_{ij}[n], Y_{ij}[n]$, $T_{ij}[n]$ y $M_{ij}[n]$ son de las mismas dimensiones que la imagen y fueron inicializadas en 0.

Los otros parámetros fueron empíricamente seleccionados para simulación como:

- La matriz interna de pesos formada con valores gaussianos en función de la distancia:

$$w_{ijkl} = \begin{bmatrix} 0.5 & 1 & 0.5 \\ 1 & 0 & 1 \\ 0.5 & 1 & 0.5 \end{bmatrix}.$$

- Constantes: $f = 0.9, g = 0.8, h = 20$.

- En el paso 4, el operador de mediana y el operador promedio están definidos de acuerdo a lo siguiente: Si x_{ij}denota al pixel con coordenadas (i, j) en la imagen con ruido y X_{ij} denota el conjunto de pixeles en la ventana W con el vecindario $(2K + 1) \times (2K + 1)$ centrada en x_{ij}, entonces:

$$X_{ij} = \{x_{i-K,j-K}, \dots, x_{ij}, \dots, x_{i+K,j+K}\}. \quad (7)$$

- La mediana de la ventana de la imagen está definida como

$$m_{ij} = mediana\ (X_{ij}). \quad (8)$$

- El promedio de la ventana de la imagen está definido como

$$p_{ij} = promedio(X_{ij}). \quad (9)$$

El algoritmo ICM-TM se comparó contra dos filtros clásicos, el filtro de mediana de 3x3 y el filtro *Wiener* de 3x3, y contra el PCNNNI basado en el algoritmo de Ma [15].

Se realizó la medición de los resultados obtenidos con tres métricas, las cuales están formuladas como sigue [13]:

a) $PSNR$ (*Peak Signal to Noise Ratio*, dB), el cual es utilizado para medir la habilidad de supresión del ruido, mientras más grande es su valor mejor es el efecto del filtrado,

$$PSNR = 10 \log_{10} \left(\frac{f(m,n)^2}{\frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} [f(m,n) - f'(m,n)]^2} \right) [dB]. \quad (10)$$

b) MAE (*Mean Absolute Error*), indica la calidad del filtrado como la preservación de detalles finos, para lo que cual de ser minimizado,

$$MAE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |f(m,n) - f'(m,n)|, \quad (11)$$

donde M y N denotan las filas y columnas de la imagen, $f(m,n)$ denota la imagen sin ruido y $f'(m,n)$ es la imagen resultante del proceso de filtrado.

c) $NMSE$ (*Normalized Mean Square Error*), un mejor método de filtrado debe generar un menor valor resultante de $NMSE$.

$$NMSE = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[f(m,n) - f'(m,n)]^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}[f(m,n)]^2}. \quad (12)$$

De manera adicional también se midió el tiempo de cómputo, el cual se realizó bajo el sistema operativo Windows 7 Ultimate SP1, con un procesador Intel Core i7 a 3.40 GHz y 8GB de RAM utilizando Matlab.

6. Desempeño del método propuesto

Para ICM cada salida es distinta en cada iteración y la activación de las neuronas obedece a la relación entre los niveles de gris de la imagen con ruido gaussiano de promedio 0 y varianza 0.01 (Fig. 3). La red puede iterarse cuantas veces se desee dependiendo del objetivo que se persiga. Para el caso de filtrado de ruido gaussiano se requiere encontrar información sobre el grado de contaminación de cada pixel, por lo que se analizó la matriz de tiempos.

La información de la matriz de tiempos es numérica, por lo que se pueden distinguir los pixeles que se activan por primera vez en cada iteración y se extraen los pixeles que guardan mayor información de la imagen, discriminando los que no para eliminar la mayor cantidad de ruido no deseado en la imagen.

a) b) c) d)

Fig. 3. Salidas binarias de la red ICM para las iteraciones a) 5, b) 6, c) 7 y d) 8

Para dar un ejemplo numérico del procedimiento de filtrado considérese una matriz correspondiente a una imagen en escala de grises contaminada con ruido gaussiano de promedio 0 y varianza 0.08 de la que se tomó una ventana de 6 x6 pixeles O_{ij} y su correspondiente imagen normalizada S_{ij}:

$$O_{ij} = \begin{pmatrix} 238 & 139 & 255 & 239 & 112 & 182 \\ 68 & 235 & 59 & 255 & 178 & 42 \\ 179 & 52 & 138 & 227 & 116 & 81 \\ 238 & 45 & 114 & 67 & 205 & 51 \\ 147 & 54 & 148 & 128 & 37 & 255 \\ 180 & 0 & 141 & 25 & 110 & 56 \end{pmatrix}, \; S_{ij} = \begin{pmatrix} 0.9333 & 0.5451 & 1.0000 & 0.9373 & 0.4392 & 0.7137 \\ 0.2667 & 0.9216 & 0.2314 & 1.0000 & 0.6980 & 0.1647 \\ 0.7020 & 0.2039 & 0.5412 & 0.8902 & 0.4549 & 0.3176 \\ 0.9333 & 0.1765 & 0.4471 & 0.2627 & 0.8039 & 0.2000 \\ 0.5765 & 0.2118 & 0.5804 & 0.5020 & 0.1451 & 1.0000 \\ 0.7059 & 0 & 0.5529 & 0.0980 & 0.4314 & 0.2196 \end{pmatrix}$$

Se introduce la imagen normalizada en la red ICM y se itera 10 veces, obteniéndose el umbral dinámico de dicha iteración T_{ij}, los valores en x son indistintos para este ejemplo, tal es el caso de los bordes de la imagen:

$$T_{ij} = \begin{pmatrix} 16.1554 & 19.3554 & 16.1554 & 16.1554 & 19.3554 & x \\ 19.3554 & 13.5954 & 16.1554 & 13.5954 & 16.1554 & x \\ 19.3554 & 19.3554 & 16.1554 & 16.1554 & 19.3554 & x \\ 16.1554 & 19.3554 & 19.3554 & 19.3554 & 16.1554 & x \\ 19.3554 & 23.3554 & 19.3554 & 19.3554 & 23.3554 & x \\ x & x & x & x & x & x \end{pmatrix}.$$

Luego se obtiene el potencial interno de la matriz F_{ij}:

$$F_{ij} = \begin{pmatrix} 9.3197 & 8.4762 & 11.4841 & 11.0754 & 5.9214 & x \\ 6.7977 & 13.7602 & 8.8778 & 14.0460 & 9.6073 & x \\ 9.7184 & 8.9600 & 11.1566 & 13.3848 & 7.9292 & x \\ 11.4509 & 9.0972 & 10.8096 & 9.5191 & 10.6130 & x \\ 7.0908 & 6.6561 & 9.2021 & 8.6462 & 4.1812 & x \\ x & x & x & x & x & x \end{pmatrix}.$$

Si el potencial interno de la neurona (ij) supera al valor de umbral, entonces se produce una salida en 1 para esa neurona; al terminar de procesar la imagen se tiene una matriz binaria de salidas Y_{ij}. El número de iteración en el que sucede la primera activación de cada neurona se guarda en la matriz de tiempos final M_{ij}.

$$Y_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \; M_{ij} = \begin{pmatrix} 7 & 8 & 7 & 7 & 8 & x \\ 8 & 6 & 7 & 6 & 7 & x \\ 8 & 8 & 7 & 7 & 8 & x \\ 7 & 8 & 8 & 8 & 7 & x \\ 8 & 9 & 8 & 8 & 9 & x \\ x & x & x & x & x & x \end{pmatrix}.$$

La etapa de filtrado selectivo consiste en recorrer las matrices calculando la mediana de los valores de la matriz de tiempos en una ventana de 5x5 denotado como $mediana\ M_{2,2}(5x5)$ y comparar el valor con el correspondiente al pixel central $M_{2,2}$ de la ventana considerada, cuyos valores respectivos son:

$$mediana\ M_{2,2}(5x5) = 8,$$

$$M_{2,2} = 7.$$

Si estas cifras son iguales el valor del pixel correspondiente en la imagen filtrada $Sfiltrada_{2,2}$ será el promedio de los valores en esta ventana en la imagen ruidosa $promedio\ O_{2,2}(5x5)$, en caso contrario el pixel correspondiente tomará el valor de la mediana de la vecindad de 5x5 de la imagen con ruido. En este caso no son iguales, se procede a calcular la mediana de la ventana en la imagen con ruido $mediana\ O_{2,2}(5x5)$ y asignarlo al pixel de la imagen filtrada $Sfiltrada_{2,2}$. El proceso termina cuando se ha procesado la imagen completa.

$$mediana\ O_{2,2}(5x5) = 138,$$

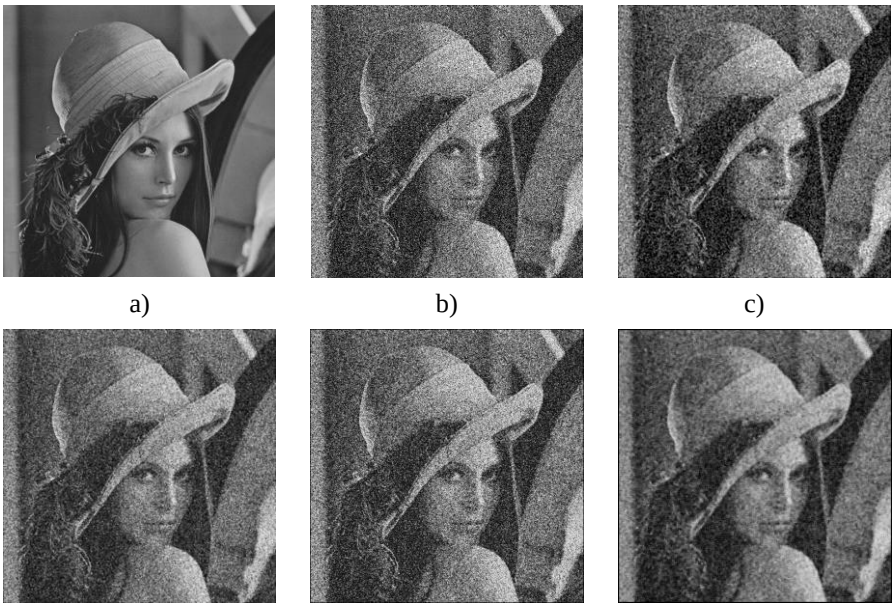
$$Sfiltrada_{2,2} = 138.$$

Se realizó la comparación del método propuesto con los filtros de mediana, *Wiener* y el basado en PCNNNI (Fig. 4); la capacidad de suavizado del filtro de mediana puede ser empleado para disminuir el ruido gaussiano que genera valores extremos en los pixeles, no obstante no se debe aplicar este filtro de manera uniforme pues la imagen

se verá afectada en detalles y bordes, mientras que para otras regiones se debe tomar en cuenta la información presente en los pixeles por medio de un suavizado más fino como es el generado por el operador promedio.

El análisis cualitativo de la imagen ampliada hace evidente que el filtro *Wiener* mantiene un alto grado de ruido aunque preserva los bordes (Fig. 5).

Por otro lado la técnica basada en PCNNNI conserva los detalles finos, pero su capacidad de supresión del ruido no es efectiva (Fig. 6).



a) b) c)

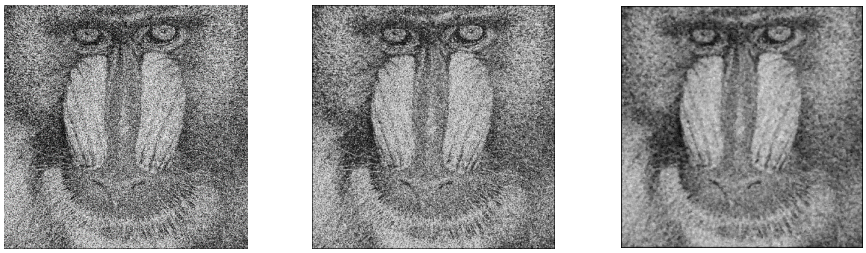
d) e) f)

Fig. 4. Comparación entre técnicas de filtrado de ruido gaussiano a) original sin ruido, b) con ruido gaussiano var. 0.08, c) mediana, d) *Wiener*, e) PCNNNI y f) ICM-TM

El análisis cuantitativo del PSNR (Fig. 7), permite mostrar que el desempeño del ICM-TM es superior por 2dB a los métodos tradicionales de filtrado y a la técnica basada en PCNNNI para imágenes donde el ruido gaussiano tiene varianza 0.07, mostrando este buen desempeño cuando la varianza del ruido es más baja, hasta 0.01 cuando el ICM-TM supera a los métodos clásicos por 1dB. Con varianzas menores el desempeño de la técnica ya no es superior.

a) b) c)

Fig. 5. a) Imagen con ruido gaussiano var. 0.08, b) filtrada con *Wiene*r, c) filtrada con ICM-TM



a) b) c)

Fig. 6. a) Imagen con ruido gaussiano var. 0.09, b) filtrada con PCNNNI y c) con ICM-TM

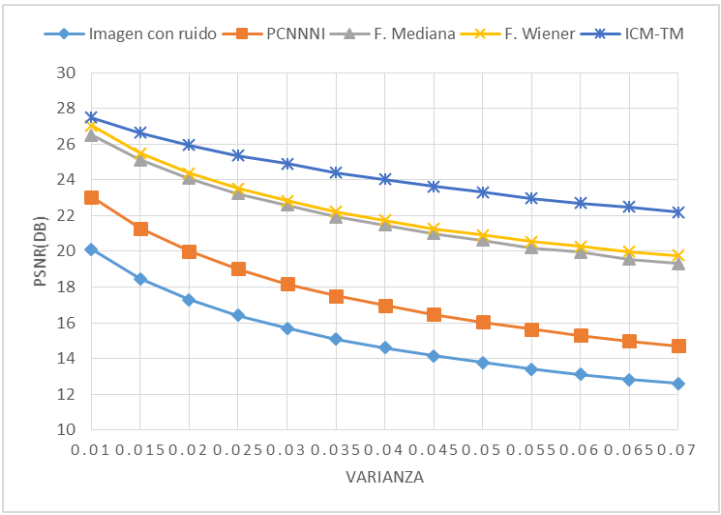


Fig. 7. Curvas de desempeño en PSNR de los filtros de ruido Gaussiano

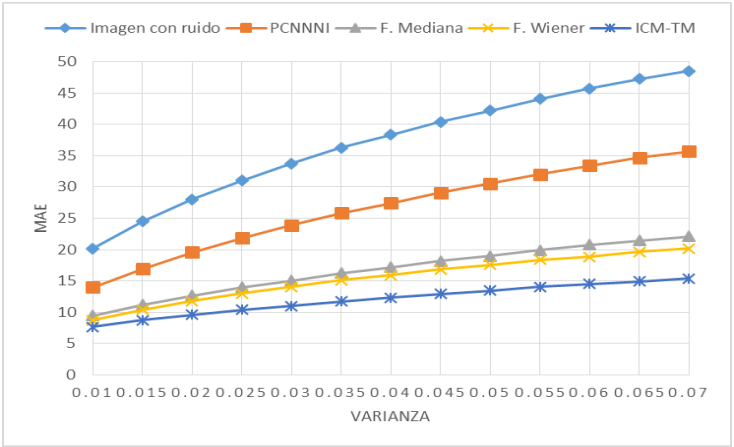


Fig. 8. Curvas de desempeño en MAE de los filtros de ruido Gaussiano

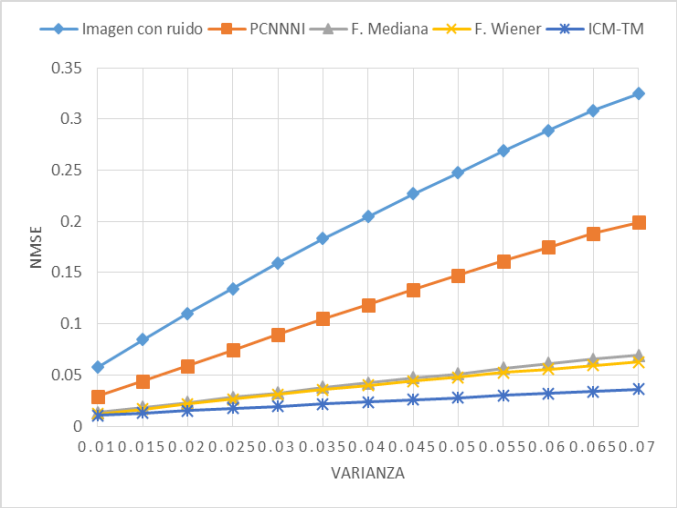


Fig. 9. Curvas de desempeño en NMSE de los filtros de ruido Gaussiano

En cuanto al análisis de los datos reflejados en el MAE se observa que la mejora de la imagen en cuanto a detalles finos es de al menos 30 puntos con respecto a la imagen original en el mejor de los casos y por 10 puntos en el peor (Fig. 8).

Mediante el análisis del NMSE (Fig. 9), se observa que ICM-TM tiende a minimizar esta métrica, lo cual muestra que es un mejor método de filtrado, seguido por el filtro *Wiener* y el filtro de mediana, donde la diferencia con el método de PCNNNI es notable por 0.3 en el mejor de los casos. Conforme la varianza del ruido aumenta el error de la imagen contaminada y la capacidad de recuperación de los filtros disminuye.

Tabla 1. Desempeño de los filtros de ruido Gaussiano

Imagen con ruido Gaussiano var.	**Imagen original**	**PCNNNI**		**F. Mediana**		**F. *Wiener***		**ICM-TM**	
	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR
Lena 0.08	12.15	0.224	14.22	0.08	18.7	0.069	19.31	0.041	**21.68**
Lena 0.09	11.8	0.245	13.83	0.088	18.29	0.075	18.97	0.044	**21.36**
Baboon 0.08	11.93	0.143	13.9	0.064	17.39	0.05	18.47	0.045	**18.9**
Baboon 0.09	11.58	0.155	13.53	0.069	17.05	0.053	18.21	0.048	**18.66**
Peppers 0.08	12.15	0.14	14.27	0.05	18.76	0.044	19.31	0.024	**21.97**
Peppers 0.09	11.76	0.154	13.85	0.056	18.26	0.048	18.91	0.027	**21.52**

En cuanto al tiempo el promedio para PCNNNI fue de 160s, para el filtro de Mediana de 0.028s, para el filtro *Wiener* de 0.036s y para ICM-TM de 20s, por lo que la complejidad del método es acorde con el tiempo de procesamiento aunque éste puede ser mejorado mediante procesamiento en paralelo.

En general para varianzas de ruido gaussiano mayores a 0.01 el desempeño de ICM-TM es superior a PCNNNI, al filtro de mediana y al filtro *Wiener* para distintas

imágenes contaminadas con ruido gaussiano como muestran los datos de NMSE y PSNR de la Tabla 1 para distintos valores de ruido gaussiano con diversas imágenes.

7. Conclusiones y trabajo futuro

Los resultados preliminares del método de filtrado ICM-TM son que supera los filtros tradicionales como filtro *Wiener* en promedio por 1.82 dB según PSNR y en 14% en NMSE y MAE, además de que relaciona una vecindad de pixeles y no requiere cálculos estadísticos ni complejos, del mismo modo aprovecha las características de los filtros promedio y mediana al aplicarlos selectivamente.

Por otro lado ICM-TM supera a PCNNNI [14] en promedio por 6.75dB, y requiere 80% menos tiempo de procesamiento, además no se realiza un cambio del rango dinámico de la imagen, se considera la relación entre neuronas vecinas y se tienen que ajustar menos parámetros.

El trabajo futuro se centrará en la exploración de PCNN con variantes y la elección de los coeficientes de ajuste por medio de algoritmos evolutivos, así como en la búsqueda de un operador que permita un mejor ajuste de los valores de la imagen para preservar mejor los detalles, el estudio profundo de la matriz de tiempos y tomando en cuenta las frecuencias para la localización de los pixeles que guardan mayor información de la imagen discriminando aquellos que presentan mayor grado de ruido. También se pretende estudiar el comportamiento del filtro como sistema dinámico y su desempeño sobre algunas imágenes artificiales que se enfoca a características específicas como bordes y líneas. Comparar su desempeño con otro tipo de filtros como el direccional o filtros difusos y ampliar el conjunto de imágenes de prueba.

Agradecimientos. Los autores agradecen al CENIDET y al IPN por el apoyo económico para la realización de la presente investigación con ayuda de los siguientes fondos: SIP-IPN 20161126, y CONACYT en el marco de los proyectos 155014 de la convocatoria de Investigación básica y 65 en el marco de la convocatoria de Fronteras de la Ciencia 2015. Estela Ortiz agradece al CONACYT por la beca concedida para la realización de sus estudios de maestría.

Referencias

1. Bovik, A.: The Essential Guide to Image Processing. Image. Elsevier Inc. (2009)
2. González, R.C.: Digital Image Processing. 2nd Ed. Prentice Hall (2002)
3. Kaur, D.: Remove Noise Effects From Degraded Document Images Using Matlab Algorithm. International Journal Of Engineering Sciences & Research Technology, Vol. 4, No. 9, pp. 544–549 (2015)
4. Kang, D., Lim, H.: Efficient noise reduction in images using directional modified sigma filter. In: Springer Science Business Media, New York, pp. 580–592 (2013)
5. Panetta, K., Bao, L., Agaian, S.: Sequence-to-Sequence Similarity Based Filter for Image Denoising. IEEE Sensors Journal (2016)
6. Liu, J., Wang, Y., Su, K., He, W.: Image denoising with multidirectional shrinkage in directionlet domain. Signal Processing, Vol. 125, pp. 64–78 (2016)

7. Kumar A., Singh, B.: Alexander Fractional Integral Filtering Of Wavelet Coefficients For Image Denoising. Signal & Image Processing : An International Journal (SIPIJ), Vol. 6, No. 3, pp. 43–54 (2015)
8. Lindblad, T., Kinser, J.M.: Image processing using pulse-coupled neural networks. Springer, 2nd Ed. (2005)
9. Eckhorn R., Reitboeck, H.J., Arndt, M.: Feature linking via synchronization among distributed assemblies: simulation of results from cat cortex. Neural Computation, pp. 293–307 (1990)
10. Ranganath, H.S., Kuntimad, G., Johnson, J.L.: Pulse coupled neural networks for image processing. In: Proceedings of IEEE Southeast Conference, Raleigh, pp. 26–29 (1995)
11. Ma, Y.D., Shi, F., Li, L.: Gaussian noise filter based on PCNN. In: Proceedings of 2003 International Conference on Neural Networks and Signal Processing, Nanjing, pp. 14–17 (2003)
12. Ma, Y.D., Lin, D.M., Zhang, B.D.: A novel algorithm of image Gaussian noise filtering based on PCNN time matrix. In: Proceedings of IEEE International Conference on Signal Processing and Communication, Dubai, pp. 24–27 (2007)
13. Lui, C., Zhang, Z.: Sonar images de-noising based on pulse coupled neural networks. In: Congress on Image and Signal Processing, pp. 403–406 (2008)
14. Yuan-yuan, C., Hai-yan, L., Xin-ling, S., Jian-hua, C.: A new method of denoising mixed noise using Limited Grayscale Pulsed Couple Neural Network. In: Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, pp. 1410–1413 (2011)
15. Ma, Y.D., Zhan, K., Wang, Z.: Applications of pulse-coupled neural networks. Springer, pp. 11–26 (2010)
16. Johnson, J.L., Padgett, M.L.: PCNN model and applications. IEEE Transactions on Neural Networks, Vol. 10, pp. 480–498 (1999)

Preprocesamiento de imágenes dermatoscopicas para extracción de características

Miguel A. Castillo Martínez, Francisco J. Gallegos Funes,
Alberto J. Rosales Silva, Rosa I. Ramos Arredondo

Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
Sección de Estudios de Posgrado, Ciudad de México,
México

castillo.m.miguel.a@gmail.com, fcogf@hotmail.com, arosaless23@gmail.com,
alesija@gmail.com

Resumen. Realizar un preprocesamiento en imágenes para su acondicionamiento y la extracción de características requiere un esquema de trabajo donde los elementos que afecten la obtención de la región de interés y su posterior procesamiento sean removidos de la imagen. Se hace una propuesta de trabajo para el procesamiento y análisis de forma de imágenes dermatoscopicas con el fin de obtener datos que caractericen la imagen y usar esta información para un diagnóstico clínico. Se utilizaron 200 imágenes dermatoscopicas para realizar un análisis de similitud entre las imágenes con y sin procesamiento, se obtuvo una similitud de 0.96±0.02 con lo cual las imágenes no son alteradas de manera que afecten, en gran medida, las siguientes etapas. Al obtener la región de interés se obtuvo un error de segmentación de 30.12±19.19% debido a una comparación entre la segmentación subjetiva y la objetiva donde, el médico, al encontrar zonas que no son parte de la lesión son asignadas a esta por estar contenidas en la lesión.

Palabras clave: Segmentación, procesamiento de imágenes, extracción de características, imágenes dermatoscopicas.

Dermoscopic Image Preprocessing for Feature Extraction

Abstract. Perform an image preprocessing for its conditioning and feature extraction needs a work scheme where the elements, that affects the region of interest and processing, are removed of the image. There is a dermoscopic image processing and shape analysis approach, with getting data that characterizes the image and use this information for clinic diagnosis. 200 dermoscopic images were used in similar analysis between the images with processing and without processing; a 0.96±0.02 similar measure was obtained showing that an image significant damage is not present for next processing stages. A segmentation error of 30.12±19.19% was obtained because a comparative between subjective and

objective segmentations where the medic classifies regions of skin that no belongs to the lesion but are contained in that.

Keywords: Segmentation, image processing, feature extraction, dermoscopic images.

1. Introducción

Un problema observado en la evaluación objetiva de imágenes es la presencia de elementos que tienen efectos negativos sobre su análisis, algunos de los factores que pueden afectar la evaluación de estas imágenes es la presencia de vello y/o burbujas por lo que se busca eliminar estos elementos de la imagen, además de que se requiere una ampliación del área a evaluar para identificar características que no se aprecian a simple vista. Para la adquisición de este tipo de imágenes se requiere un dermatoscopio el cual cuenta con una lente de magnificación y una fuente de luz para realizar el diagnostico de lesiones en la piel [1]. Otro de los factores es la naturaleza de las imágenes, toda imagen contiene variaciones de iluminación, color y textura por lo que se dice que son ruidosas requiriendo una homogeneización sin perder detalles importantes que ayuden a un mejor análisis de la imagen obtenida al procesar la imagen para su caracterización.

Para caracterizar una imagen correctamente se requiere obtener una región de interés que pueda describir la lesión contenida en la misma, esto representa una tarea difícil para médicos poco experimentados y puede variar por criterios tomados al momento de hacer esta clasificación, además de la correcta evaluación de la lesión respecto a estos criterios. Se requiere el desarrollo de herramientas computacionales que asistan en la caracterización de imágenes dermatoscopicas adquiridas de distintos medios y en condiciones poco ideales.

En el presente trabajo se propone un esquema que ayude al acondicionamiento de la imagen para ser evaluada medicamente y finalizar con la extracción de características de la misma utilizando esta información para determinar el tipo de lesión presente, donde una gran alteración de la imagen no es deseable debido a que puede eliminar información importante de la lesión pero se debe modificar parte de ella para evitar dar un diagnostico alejado a lo esperado.

2. Trabajos relacionados

En [2] se utiliza una metodología para la extracción de colores donde su pre-procesamiento consiste en la binarización de la imagen seguida de un filtro de máximo para la eliminación de vellos, etiquetas o marcas de la imagen que se adquirió para tener una región de interés de la lesión, posteriormente se hace una difusión lineal para hacer más homogénea la imagen y finalizando con la clasificación de la información para obtener los colores. En [3] se considera un cambio a escala de grises continuando con la eliminación de esquinas y un filtrado de mediana para remover vello y burbujas pero se compromete la definición de la imagen y finalmente en [4] se propone el uso de algoritmos de detección de vello para el retoque de la imagen eliminando elementos

bien definidos de la imagen como el vello y burbujas sin comprometer la definición y conservando patrones que puedan proporcionar información adicional de las imágenes.

3. Métodos y materiales

En esta sección se presenta la propuesta de trabajo así como los métodos utilizados para el procesamiento de las imágenes dermatoscopicas y la extracción de características.

En Fig. 1 se ilustra el diagrama de bloques para el procesamiento de imágenes dermatoscopicas para la obtención de la región de interés, dando la localización de la lesión en la imagen de trabajo donde al finalizar se hace la extracción de características realizando un análisis de forma basada en el análisis de momentos.

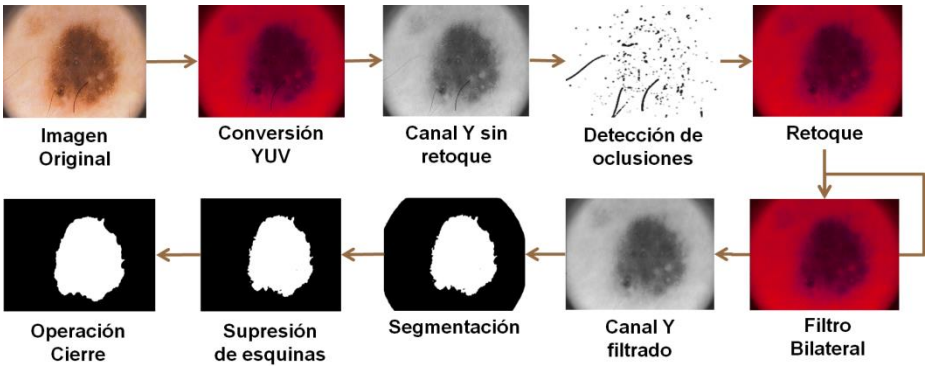


Fig. 1. Diagrama de bloques la propuesta de trabajo

El esquema consiste en tomar la imagen y realizar una conversión de espacio de color, en este caso, de RGB a YUV. Al tener nuestra imagen en el nuevo espacio de color se toma el canal de luminancia y se procede a la detección de oclusiones para obtener la máscara de los pixeles que serán procesados en el retoque. La etapa de retoque utiliza la imagen en el nuevo espacio de color y, basado en la máscara de oclusiones, se hace el retoque de la imagen con el fin de eliminar vello y burbujas presentes que puedan afectar en la identificación de la lesión.

Una imagen, por la gran cantidad de cambios de contraste, textura o color que puede contener, se puede considerar de naturaleza ruidosa, es cuando se decide hacer un filtrado bilateral para disminuir las variaciones sin perder los detalles de la imagen pasando a una etapa de segmentación donde se pretende clasificar los pixeles que pertenecen a la lesión. Cuando se adquiere la imagen se acompaña de la presencia de cuatro esquinas que pueden afectar la visualización causadas por el acoplamiento de las lentes al sistema de adquisición de imágenes, por lo que se requiere que los datos en estas esquinas sean discriminados con una etapa de supresión de datos o regiones. Se utiliza una última etapa donde se busca eliminar regiones de interés muertas que se encuentren dentro de la lesión para proceder con la extracción de las características que estén presentes en las imágenes y estos datos puedan ser usados para una clasificación.

a. Imágenes dermatoscopicas

Se utiliza la base de datos PH2, la cual cuenta con 200 imágenes dermatoscopicas adquiridas con una magnificación de 20x, una resolución de 768x560 pixeles y luz polarizada [5], además incluye la segmentación médica de la lesión, diagnóstico clínico e histológico y la evaluación de algunos criterios dermatológicos.

b. Espacio de color YUV

Comienza con colores RGB y asume que una luz blanca D65 fue utilizada para adquirir la imagen. Es usado en codificaciones de color para la transmisión de señales analógicas de televisión. Este espacio separa las componentes cromáticas por medio de la substracción de iluminación Y de los canales rojo y azul respectivamente [6]. La Matriz de transformación asociada es la siguiente:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147141 & -0.288869 & 0.436010 \\ 0.614975 & -0.514965 & -0.100010 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (1)$$

Es un enfoque de espacio de color ampliamente usado en el desarrollo de algoritmos de procesamiento de imágenes a escala de grises.

c. Detección de oclusiones

Consideramos por oclusión a todo elemento en la imagen que afecte o tenga poca importancia para el procesamiento de la misma. La detección de oclusiones se realiza por medio de la desviación estándar, tomando como base el canal de luminancia en la imagen, en un pixel con una vecindad de *NxN* obteniendo una máscara de desviaciones en la cual se aplicara un umbral T_h obteniendo al final de este proceso una máscara indicando los pixeles que deberán ser retocados (Fig. 2).

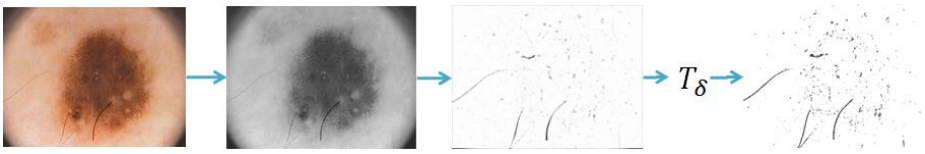


Fig. 2. Proceso de la detección de oclusiones a partir del canal de luminancia en la imagen, Obtención de la máscara de desviaciones y su umbralización para la obtención de los pixeles a retocar

d. Retoque

El retoque es una técnica que consiste en el relleno de regiones de datos perdidas, dañadas o que se busque modificar para obtener un efecto deseado. Sus aplicaciones van desde la restauración, eliminación de elementos no deseados e incluso efectos especiales [7].

En este caso se usa la imagen original y una máscara en la que se muestran los elementos a retocar, como el vello o burbujas (Fig. 3).

Fig. 3. Proceso de retoque a) Imagen original, b) Mascara de retoque, c) Imagen con retoque

Retoque basado en convolución. La convolución establece una conexión entre los dominios espacial y frecuencial, es usada para lograr una gran variedad de efectos [8]. Utiliza una matriz cuadrada de coeficientes para determinar que elementos de una imagen están en la vecindad de un pixel en particular además de especificar como la vecindad afecta el resultado y cada resultado es una combinación ponderada de la vecindad. Es un proceso lineal donde la suma y multiplicación son las operaciones aritméticas involucradas, como se muestra en la ecuación siguiente.

$$I'(x,y) = \sum_{j=-\frac{M}{2}}^{\frac{M}{2}} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} K(i,j)I(x-j,y-k). \tag{2}$$

Se hace uso de una máscara de convolución en la cual se le da un peso específico a los pixeles que comparten la ventana, en la cual el dato discriminado es el pixel a estimar.

En este caso se usa el enfoque de Hadhoud [9] el cual usa una máscara en la cual se le da un peso nulo al pixel inferior derecho.

e. Filtro bilateral

El filtro bilateral difumina imágenes mientras conserva los bordes por medio de una combinación no lineal de valores en la imagen [10]. El método es local, simple y puede ser iterativo, combina colores basados en su cercanía geométrica y similitud fotométrica teniendo más ponderación valores cercanos que lejanos en ambos dominios. La imagen filtrada es obtenida por la siguiente ecuación:

$$x_i' = \frac{\sum_{j \in N_i} \omega(i,j) x_j}{\sum_{j \in N_i} \omega(i,j)}, \tag{3}$$

donde ω son ponderaciones aplicadas a cada pixel x_j en la vecindad N_i. La ponderación se compone de la ponderación espacial ω_s y la ponderación de color ω_c.

$$\omega(i,j) = \omega_s(i,j) \times \omega_c(i,j), \tag{4}$$

donde ω_s y ω_c se definen por

$$\omega_s(i,j) = e^{\left(\frac{d_s(i,j)}{2{\sigma_s}^2}\right)}, \tag{5}$$

$$\omega_c(i,j) = e^{\left(\frac{d_c(i,j)}{2{\sigma_c}^2}\right)}, \tag{6}$$

Teniendo que σ_s está relacionado con el radio de difuminado, si σ_s es alto el difuminado es mayor pero si σ_s es muy grande puede difuminar bordes importantes. Y σ_c determina cuanto contraste será conservado o difuminado. Para valores bajos de σ_c casi todos los contrastes serán conservados mientras que para valores altos el comportamiento sería un emborronamiento gaussiano lineal [11].

f. Fuzzy C Means

Es un algoritmo de clasificación que permite a un dato pertenecer a dos o más grupos con diferente grado de pertenencia. Dado un conjunto de datos $X = \{x_1, x_2, \ldots, x_k\}$ conteniendo n número de datos de dimensión d, Fuzzy C Means pretende minimizar la siguiente función objetivo [12].

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} {u_{ik}}^m \|x_k - v_i\|^2, \tag{7}$$

donde v es el centroide del grupo i, u_{ik} es el grado de pertenencia de x_k en el grupo i mientras que m es el factor de fuzzyficación. La minimización se obtiene mediante actualizaciones sucesivas de v y u_{ik} utilizando

$$v_i = \frac{\sum_{k=1}^{n} {u_{ik}}^m x_k}{\sum_{k=1}^{n} {u_{ik}}^m}, \tag{8}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{2/(m-1)}}. \tag{9}$$

g. Supresión de esquinas

Como se muestra en Fig. 4, para suprimir las esquinas en la segmentación se considera que, por efecto del acoplamiento de las lentes, la lesión está dentro de una circunferencia de radio r por lo que las regiones son discriminadas si no se encuentran dentro de esta circunferencia concéntrica a la imagen.

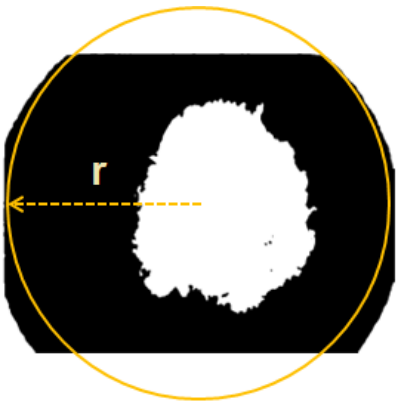


Fig. 4. Enfoque para la supresión de esquinas de la segmentación

h. Extracción de características

La asimetría y la longitud de los ejes son características a extraer de las imágenes a partir de su segmentación cuyos procedimientos se detallan a continuación.

Momentos. La ecuación 10 describe el momento del orden p, q para una imagen $I(x, y)$ que pertenece a la región de interés $\mathcal{R}$ como puede ser una imagen a escala de grises [13].

$$m_{pq} = \sum_{(x,y)\in\mathcal{R}} I(x,y)x^p y^q. \tag{10}$$

En el caso de una región binaria $I(x, y) \in \{0,1\}$, solo los pixeles de primer plano con $I(x, y) = 1$ en la región necesitan ser considerados, por lo tanto la ecuación Y puede ser simplificada a

$$m_{pq} = \sum_{(x,y)\in\mathcal{R}} x^p y^q. \tag{11}$$

De este modo, el área de una región binaria puede ser expresada como un momento de orden 0

$$A = \sum_{(x,y)\in\mathcal{R}} 1 = \sum_{(x,y)\in\mathcal{R}} x^0 y^0 = m_{00.} \tag{12}$$

Y del mismo modo el centroide $\bar{c}$ como

$$\bar{c} = (\bar{x}, \bar{y}) = \left(\frac{\sum_{(x,y)\in\mathcal{R}} x^1 y^0}{\sum_{(x,y)\in\mathcal{R}} x^0 y^0}, \frac{\sum_{(x,y)\in\mathcal{R}} x^0 y^1}{\sum_{(x,y)\in\mathcal{R}} x^0 y^0}\right) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right). \tag{13}$$

Los momentos representan propiedades físicas de la región. Específicamente, el área es una base importante en la práctica para la caracterización de regiones, y el centroide permite la confiable y exacta especificación de la posición de la región con una fracción de pixel de error.

Momentos centrales. Para calcular las características invariantes a posición se puede usar el centroide de la región como punto de referencia. En otras palabras, podemos mover el origen del sistema coordenado al centroide $\bar{c}$ para obtener los momentos centrales de orden p, q

$$\mu_{pq} = \sum_{(x,y)\in\mathcal{R}} I(x,y)(x-\bar{x})^p (y-\bar{y})^q. \tag{14}$$

Para una imagen binaria puede ser simplificada a

$$\mu_{pq} = \sum_{(x,y)\in\mathcal{R}} (x-\bar{x})^p (y-\bar{y})^q. \tag{15}$$

Excentricidad. Basado en los momentos de la región, mediciones altamente precisas y estables pueden ser obtenidas sin ninguna búsqueda iterativa u optimización. Además, los métodos basados en momentos no requieren conocer el perímetro para calcular la

circularidad, y pueden manipular regiones no conectadas o nubes de puntos. Se adopta la siguiente definición por su simple interpretación geométrica [13].

$$Ecc = \frac{a_1}{a_2} = \frac{\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4{\mu_{11}}^2}}{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4{\mu_{11}}^2}}. \tag{16}$$

La longitud de los ejes de la elipse descrita por la región está dada por

$$r_{ax} = \left(\left(\frac{2a_1}{m_{00}}\right)^{\frac{1}{2}}, \left(\frac{2a_2}{m_{00}}\right)^{\frac{1}{2}} \right). \tag{17}$$

Asimetría. El coeficiente de asimetría u oblicuidad mide el grado de asimetría en una distribución, cuanto mayor sea el coeficiente más asimétrica será la distribución. Un valor positivo indica una oblicuidad a la derecha, un valor negativo indica una oblicuidad a la izquierda y si el valor es 0 indica que la distribución es simétrica.

El cálculo de la asimetría está dado por la ecuación 18 la cual es una ecuación de tipo 2 adoptada por software estadístico como SAS, SPSS y la hoja de cálculo Excel [14].

$$G = \left(\frac{m_{30}}{{m_{20}}^{\frac{3}{2}}}, \frac{m_{03}}{{m_{02}}^{\frac{3}{2}}} \right) * \frac{\sqrt{m_{00}(m_{00} - 1)}}{(m_{00} - 2)}. \tag{18}$$

i. Cierre

Está una operación compuesta por las operaciones morfológicas dilatación y erosión respectivamente denotada por la siguiente ecuación.

$$I \bullet H = (I \oplus \mathrm{H}) \ominus H. \tag{19}$$

La dilatación es una operación que corresponde al crecimiento, donde se agregan capas de pixeles a la imagen I dependiendo de un elemento estructural H y se denota por la siguiente ecuación

$$I \oplus H \equiv \bigcup_{p \in I} H_p. \tag{20}$$

La erosión es una operación cuasi-inversa a la dilatación cuya función es remover pixeles respecto al elemento estructural H donde su notación es la siguiente

$$I \ominus H \equiv \{\mathrm{p} \in \mathbb{z}^2 | H_p \subseteq I\}, \tag{21}$$

donde *p* es un par coordenado en *I(p)=1* y H_p denota el elemento estructural desplazado por *p*.

El elemento estructural es similar a la matriz de coeficientes de un filtro lineal, las propiedades de un filtro morfológico son especificadas en la matriz de elementos *H*. En una imagen binaria *H* contendrá valores de 0 y 1 solamente, para el presente trabajo se utilizó una *H* tipo disco de 10x10.

4. Resultados

El retoque es evaluado utilizando una medición llamada Structural Similarity Index Metric (SSIM) la cual es una medición objetiva de referencia total que entrega el grado de similitud entre dos imágenes. El SSIM modela las distorsiones en una combinación de baja correlación, distorsión de luminancia y distorsión de contraste [15], para calcular el SSIM se utiliza la siguiente ecuación

$$SSIM = \frac{(2\bar{x}\bar{y} + C_1)(2\sigma_{xy} + C_2)}{(\sigma_x{}^2 + \sigma_y{}^2 + C_2)(\bar{x}^2 + \bar{y}^2 + C_1)}, \quad (22)$$

donde C_1 y C_2 Son constantes para estabilizar la operación cuando $(\bar{x}^2 + \bar{y}^2)$ o $(\sigma_x{}^2 + \sigma_y{}^2)$ son muy cercanos a cero.

Para la evaluación se utiliza la imagen original x, la imagen retocada y, C_1=*6.5* y C_2 =*58.52*.

La evaluación entrego que el SSIM=0.96±0.02 indicando que el retoque no afecta en gran medida la imagen para una siguiente etapa de procesamiento. En Fig. 5 se pueden apreciar retoques sobre 2 imágenes.

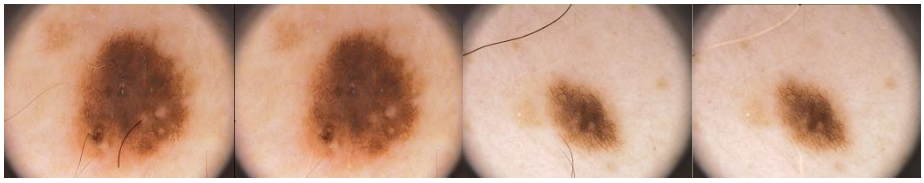


Fig. 5. Retoque sobre dos imágenes de la base de datos con un SSIM de 0.9858 y 0.9730

Este tipo de retoque no copia la textura presente en el área de retoque y si la detección de oclusiones no es la adecuada el retoque puede tener un efecto negativo sobre la imagen, esto se representa en la Fig. 6.

Fig. 6. Retoque a imagen con detección de oclusiones poco eficiente con un SSIM = 0.8899

En el caso del error de segmentación (SE) está definido con la siguiente ecuación [16]

$$SE = \frac{A(GT \oplus SR)}{A(GT)} \times 100\%, \quad (23)$$

donde *GT* es la segmentación manual de la lesión y *SR* es el resultado de la lesión. De forma similar que en el SSIM se toma la segmentación manual y la segmentación obtenida con el Fuzzy C Means, se obtuvo un SE=30.12±19.43%.

Además se evalúa razón de detección verdadera (*TDR*) y la razón de falso positivo (*FPR*) definidas en [16] con las ecuaciones 21 y 22 dando 71.96±20.05% y 2.08±6.08% respectivamente.

$$TDR = \frac{A(SR \cap GT)}{A(GT)}, \tag{24}$$

$$FPR = \frac{A(SR \cap \overline{GT})}{A(GT)}. \tag{25}$$

Estos valores se deben a la evaluación subjetiva de las imágenes donde la segmentación de la base de datos tiende a un margen de error por la evaluación objetiva de los métodos, esto se presenta en Fig. 7 comparando dos imágenes con su segmentación médica, donde se discrimina la piel y se toma como parte de la lesión, y la comparación de la segmentación automática, donde la piel es separada de la lesión.

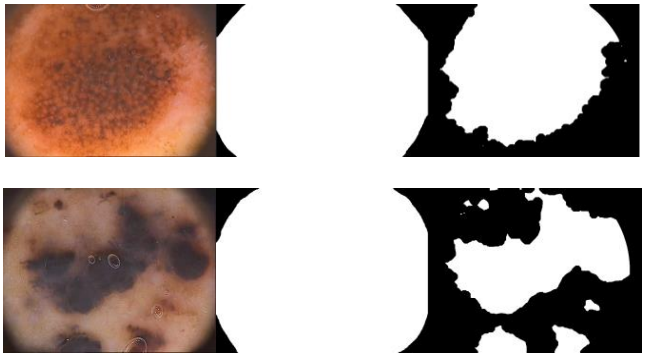


Fig. 7. Segmentación de dos imágenes dermatoscopicas donde la piel es discriminada y se toma como parte de la lesión. De izquierda a derecha: Imagen original, Segmentación médica, Segmentación automática. De arriba abajo: SE = 43.64%, SE = 59.53%

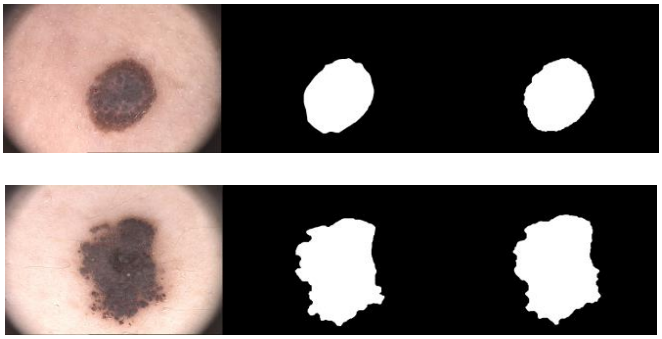


Fig. 8. Segmentación de dos imágenes dermatoscopicas donde la evaluación subjetiva no afecta. De izquierda a derecha: Imagen original, Segmentación médica, Segmentación automática. De arriba abajo: SE = 6.3%, SE = 6.55%

Haciendo una comparación con las imágenes donde la evaluación subjetiva no tenga una influencia negativa en las mediciones podemos representar con la siguiente figura los efectos de este tipo de evaluación (Fig. 8).

5. Conclusión

Se realizó una etapa de preprocesamiento para la obtención de regiones de interés en imágenes dermatoscopicas relacionadas con lesiones presentes en las mismas, además de la extracción de características para su posterior evaluación y posible clasificación. El retoque intenta eliminar las oclusiones presentes en la imagen sin alterar la imagen de manera que sea útil para continuar con el procesamiento, pero aun depende de la correcta detección de las oclusiones. La segmentación entrega resultados de acuerdo a una función objetiva, por lo que la comparación de los resultados aún dependerá de la subjetividad con la que se hacen las segmentaciones manuales, esta dependencia varía desde las condiciones en que se tomó la fotografía y los criterios de evaluación para su análisis. Es necesario optimizar la detección de oclusiones para obtener mejores resultados en el retoque y así minimizar los errores que se encuentren en su segmentación, además de ajustar los parámetros de difuminado para la correcta uniformidad de la imagen sin perder los detalles importantes y evitar un mal análisis.

Agradecimientos. Los autores agradecen a la Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica (unidad Zacatenco) del Instituto Politécnico Nacional y al Consejo Nacional de Ciencia y Tecnología de México (CONACYT) por la ayuda y el apoyo con número de proyecto 240820.

Referencias

1. Marghoob, A.A., Malvehy, J., Braun, R.P.: Atlas of Dermoscopy. Informa Healthcare, London (2012)
2. Almaraz, J.A., Ponomaryov, V.: Extracción de Colores en Imágenes Dermatoscopicas. 15vo Congreso Nacional de Ingeniería Electromecánica y de Sistemas (2015)
3. Caeiro, L.: Automatic System for Diagnosis of Skin Lesions Based on Dermoscopic Images. Dissertation to Obtain the Master Degree in Biomedical Engineering (2009)
4. Jaworek, J.: Automatic Detection of Melanomas: An Application Based on the ABCD Criteria. In: Pietka, E., Kawa, J (eds.) Information Technologies in Biomedicine 2012. LNCS, Vol. 7339, pp. 67–76. Springer, Heidelberg (2012)
5. Addi project. http://www.fc.up.pt/addi
6. Reinhard, E., Khan, E.A., Akyüz, A.O., Johnson, G.: Color Imaging: Fundamentals and Applications. A.K. Peters, Ltd., Massachusetts (2008)
7. Oliveira, M.M., Bowen, B., McKenna, R., Chang, Y.: Fast Digital Image Inpainting. In: Proceedings of the International Conference on Visualization, Imaging and Image Processing, pp. 261–266 (2001)
8. Hunt, K.A.: The Art of Image Processing with Java. A K Peters, Ltd., Massachusetts (2010)
9. Mohiy, M.H., Kamel, A.M., Sameh, Z.S.: Digital Images Inpainting using Modified Convolution Based Method. International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 1, No. 1, pp. 1–10 (2008)

10. Tomasi, C., Manduchi, R.: Bilateral Filtering for Gray and Color Images. In: IEEE International Conference on Computer Vision, pp. 839–846 (1998)
11. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-Time Video Abstraction. ACM Transactions on Graphics (TOG), Vol. 25, No. 3, pp. 1221–1226 (2006)
12. Wu, J.: Advances in K-means Clustering: A Data Mining Thinking. Springer, Berlin (2012)
13. Burger, W., Burge, M.J.: Digital Image Processing: An Algorithmic Introduction using Java. Springer, New York (2008)
14. Yusoff, S.B., Wah, Y.B.: Comparison of Conventional Measures of Skewness and Kurtosis for Small Sample Size. Faculty of Computer and Mathematical Sciences
15. Varnan, C.S., Jagan, A., Kaur, J., Jyoti, D., Dr. Rao, D.S.: Image Quality Assessment Techniques in Spatial Domain. IJCST, Vol. 2, No. 3, pp. 177–184 (2011)
16. Wong, A., Scharcanski, J., Fieguth, P.: Automatic Skin Lesion Segmentation via Iterative Stochastic Region Merging. IEEE Transactions on Information Technology in Biomedicine, Vol.15, No. 6, pp. 929–936 (2011)

Segmentación de imágenes de color imitando la percepción humana del color

Miguel Contreras Murillo, Farid García Lamont,
Alma Delia Cuevas Rasgado

Universidad Autónoma del Estado de México,
Centro Universitario UAEM Texcoco, Texcoco-Estado de México,
México

miguelc297@gmail.com, fgarcial@uaemex.mx, almadeliacuevas@gmail.com

Resumen. Normalmente, en los trabajos de segmentación de imágenes de color entrenan redes neuronales con los colores de la imagen a segmentar, después se obtiene el número de colores dominantes dentro de la imagen para posteriormente procesar la imagen con fuzzy c-means, en donde los colores son representados en el espacio RGB. La desventaja que tienen estos trabajos es que, por un lado, deben entrenar las redes neuronales cada vez que se procesa una nueva imagen; por otro lado, en el espacio RGB la cromaticidad de un color puede ser modificada por su intensidad. En este trabajo proponemos segmentar las imágenes con información cromática de los colores, entrenando una red neuronal con muestras de cromaticidad de diferentes colores, que puede emplearse para segmentar cualquier imagen sin necesidad de volverla a entrenar; la cantidad de colores que reconoce la red neuronal depende de su tamaño. Se presentan experimentos con imágenes de la base de segmentación de Berkeley empleando redes neuronales competitivas y mapas auto-organizados.

Palabras clave: Segmentación, redes neuronales artificiales, espacios de color.

Color Image Segmentation by Mimicking the Human Perception of Color

Abstract. Usually, related works on color image segmentation train neural networks with the colors of the image to segment, then the number of dominant colors within the image is obtained in order to process the image using fuzzy c-means, where the colors are represented in the RGB space. The drawback with these methods is the neural networks must be trained every time a new image is given; but also, in the RGB space the color's chromaticity can be altered by its intensity. In this paper we propose to segment the images using chromatic data of colors, by training a neural network with chromaticity samples of different colors, which can be employed to segment any image just training it just once; the number of colors the neural network recognizes depends on its size. We show experiments with images of the Berkeley segmentation database using competitive neural networks and self-organizing maps.

Keywords: Segmentation, artificial neural networks, color spaces.

1. Introducción

La segmentación de imágenes es un tema ampliamente estudiado para la extracción y reconocimiento de objetos, de acuerdo a las características de textura, color, forma, entre otros. Dependiendo de la naturaleza del problema, las características de color de los objetos pueden proporcionar información relevante sobre ellos. Por ejemplo, la segmentación de imágenes de color ha sido aplicado en diferentes áreas como análisis de alimentos [1,2], geología [3], medicina [4,5] entre otras [6-9].

Los trabajos que abordan la segmentación de imágenes por características de color emplean diferentes técnicas [10,11], pero las más empleadas son las redes neuronales (RN) [12-14] y métodos basado en agrupamiento, específicamente, fuzzy c-means (FCM) [15-20]. Las RN son entrenadas para reconocer colores específicos, es decir, estas son entrenadas con los colores de la imagen a ser segmentada. Si se da una nueva imagen la RN debe ser entrenada nuevamente. Al emplear métodos basados en agrupamiento, se crean grupos de colores con características similares. La desventaja con tales métodos es que se requiere definir previamente la cantidad de grupos en que se divide la información; por lo tanto, el número de grupos se define dependiendo de la naturaleza de la escena.

Nuestra propuesta consiste en entrenar a la RN para reconocer diferentes colores, tratando de emular la percepción humana del color. Los seres humanos identifican principalmente los colores por su cromaticidad, después por su intensidad [21]. Por ejemplo, si se le pregunta a cualquier persona cual es el color de los cuadros (a) y (b) de la Fig. 1, lo más seguro es que responderá “verde”; nótese que el cuadro (a) es más brilloso que el cuadro (b) pero la cromaticidad no cambia. Ahora, si se le vuelve a preguntar a esa misma persona cual es el color de los cuadros (c) y (d) de la Fig. 1, lo más seguro es que responda “rojo y rosa, respectivamente”; es importante mencionar que los cuadros (c) y (d) tienen la misma intensidad pero diferentes cromaticidades.

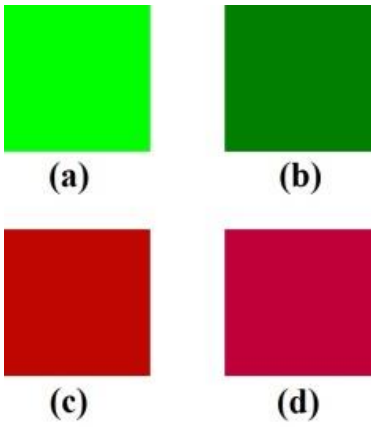


Fig. 1. Cuadros (a) y (b) con la misma cromaticidad pero con diferentes intensidades; cuadros (c) y (d) con diferentes cromaticidades pero con la misma intensidad

Los seres humanos tienen la capacidad innata de reconocer colores; al observar su entorno pueden reconocer, hasta cierto punto, regiones y/o objetos dentro de una escena por sus características cromáticas. Es importante mencionar que los humanos no necesitan aprende a identificar los colores cada vez que se les muestra una escena; ellos solo emplean su conocimiento previamente adquirido.

La contribución de este artículo es una propuesta para segmentar imágenes de color por características cromáticas, emulando la forma en que los seres humanos reconocer los colores. Con el fin de emular esta capacidad humana, proponemos entrenar un mapa auto-organizado (MAO), con muestras de cromaticidad de diferentes colores, una vez entrenada, el MAO procesa la imagen. En donde dependiendo de la cantidad de neuronas que tenga la RN es la cantidad de colores que puede reconocer la RN y en consecuencia el número de secciones que puede tener la imagen.

En la mayoría de los trabajos relacionados se emplea el espacio RGB para representar colores; sin embargo, este espacio es sensible a la iluminación por lo que la extracción de la cromaticidad de los colores no es precisa porque esta puede ser alterada por los cambios de intensidad. De aquí que, nosotros empleamos el espacio de color HSV, porque en este espacio la cromaticidad es separada de la intensidad [22].

El artículo está organizado de la siguiente forma: en la sección 2 se muestran las características de los espacios de color RGB y HSV. Presentamos nuestra propuesta para la segmentación de imágenes en la sección 3. En la sección 4 se muestran los experimentos realizados y se discuten los resultados obtenidos. Finalmente, el artículo termina con las conclusiones y trabajo futuro en la sección 5.

2. Espacios de color

Aunque el espacio RGB es ampliamente aceptado para representar colores por la comunidad de procesamiento de imágenes, los seres humanos no perciben el color como es representado en dicho espacio. La percepción humana del color es similar a la representación en el espacio HSV [21,22], de aquí que empleamos este espacio. En las secciones 2.1 y 2.2 se presentan las características de cada espacio.

2.1 Espacio de color RGB

El espacio RGB está basado en el sistema de coordenadas Cartesiano en donde los colores son puntos definidos por vectores que se extienden desde el origen, en donde el negro está en el origen y el blanco está ubicado en la esquina opuesta al origen [22], ver Fig. 2.

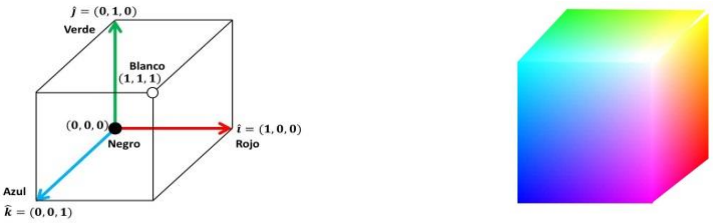


Fig. 2. Espacio de color RGB

El color de un pixel p se escribe como una combinación lineal de los vectores base de verde, rojo y azul [22]:

$$\phi_p = r_p\hat{\imath} + g_p\hat{\jmath} + b_p\hat{k}. \quad (1)$$

en donde r_p, g_p y b_p son los componentes de rojo, verde y azul, respectivamente. La orientación y magnitud de un vector de color define la cromaticidad e intensidad del color, respectivamente [22]. Como se ha mencionado antes, este espacio es sensible a la iluminación; es decir, a pesar de que dos vectores tengan la misma cromaticidad, estos representan diferentes colores si sus intensidades son diferentes.

2.2 Espacio de color HSV

La representación de color en el espacio HSV emula la percepción humana del color ya que la cromaticidad es desacoplada de la intensidad [21,22]. En este espacio el color de un pixel p se representa por sus componentes de tono (h), saturación (s) e intensidad (v):

$$\varphi_p = [h_p, s_p, v_p]. \quad (2)$$

El tono es la cromaticidad, la saturación es el nivel de blancura del color y la intensidad es el brillo del color; la Fig. 3 muestra la apariencia del espacio HSV. Los rangos de valores reales del tono, saturación e intensidad son $[0,2\pi]$, $[0,1]$ y $[0,255]$, respectivamente.

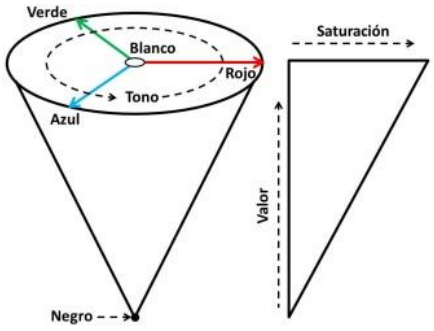


Fig. 3. Espacio de color HSV

3. Propuesta de segmentación

En esta sección presentamos nuestra propuesta para la segmentación de imágenes de color, en donde entrenamos una RN con muestras de cromaticidad de diferentes colores. Posteriormente se extrae la cromaticidad de cada pixel de la imagen a segmentar al mapear el color al espacio HSV, la cromaticidad extraída es procesada por la RN y el nuevo tono del pixel es el tono de la neurona ganadora. Cuando todos los pixeles son procesados, la imagen es mapeada al espacio RGB.

3.1 Entrenamiento de la red neuronal

Debido a la naturaleza difusa del color, no es posible reconocer todos los colores del espectro; de aquí que, el espectro de color es "dividido" en un conjunto finito de colores. El número de colores que la RN puede reconocer depende de su tamaño; en este artículo se realizan pruebas con una red neuronal competitiva (RNC) con 9, 16 y 25 neuronas, y con MAO de 3×3, 4×4 y 5×5 neuronas. Las RN son entrenadas con los elementos del conjunto Ψ construido con muestras de cromaticidad como sigue:

$$\Psi = \left\{ \psi_k = [\cos\theta_k, \sin\theta_k] \middle| \theta_k = \frac{2\pi}{256} k : k = 0,1,\ldots,255 \right\}. \quad (3)$$

La cromaticidad es transformada en un vector debido al caso cuando el valor del tono es casi 0 o 2π. Considérese los cuadros (c) y (d) de la Fig. 1, sus valores son $\pi/100$ y $19\pi/10$, respectivamente. Numéricamente ambos valores son muy distintos pero las cromaticidades de ambos cuadros son muy similares; si la cromaticidad de ambos cuadros es clasificado solamente por el valor escalar del tono, la cromaticidad es reconocida como si fueran muy diferentes.

Este problema se resuelve como sigue; sea φ_p el color de un pixel representado en el espacio HSV como se muestra en la ec. (2), le cromaticidad es modelada como:

$$\psi_p = [\cos h_p, \sin h_p]. \quad (4)$$

3.2 Procesamiento de la imagen

La segmentación de la imagen se hace al agrupar los colores empleando la cromaticidad de los colores de cada pixel de la imagen. Es importante mencionar que las RNs son entrenadas con información de la cromaticidad de los colores, por lo que no pueden reconocer el negro ni el blanco porque estos dos colores no tienen una cromaticidad definida. El blanco se obtiene cuando la saturación de un color es bajo, es decir, cuando $s \approx 0$; por otra parte, el negro se obtiene cuando la intensidad del color es baja, esto es, cuando $v \approx 0$.

Por lo tanto, antes de que un color sea procesado por la RN se debe evaluar su saturación e intensidad para clasificarlo como blanco o negro, respectivamente. Procesar el color de un pixel conlleva realizar los siguientes pasos. Sea el vector de color ϕ_p del pixel p representado en el espacio RGB:

1. El vector ϕ_p se mapea al espacio HSV obteniendo $\varphi_p = [h_p, s_p, v_p]$.
2. Se verifica si el color del pixel es negro; si $v_p \leq \delta_v$ entonces $v_p^* = 0$ y $s_p^* = 0$, ir al paso 5.
3. En caso contrario, se verifica si el color del pixel es blanco; si $s_p \leq \delta_s$ entonces $v_p^* = 191$ y $s_p^* = 0$, ir al paso 5.
4. En caso contrario, es decir, si $v_p > \delta_v$ y $s_p > \delta_s$ entonces:
   a. Calcular el vector ψ_p y procesarlo con la RN.
   b. Se obtiene el vector de peso de la neurona ganadora $\mathbf{w}_i = [w_{i,1}, w_{i,2}]$ y se etiqueta al pixel con el número i.
   c. Calcular el tono con $h_p^* = \tan^{-1}(w_{i,2}/w_{i,1})$.

d. Se asignan los valores de saturación e intensidad: $v_p^* = 191$ y $s_p^* = 1$.

5. El nuevo vector $\varphi_p^* = [h_p^*, s_p^*, v_p^*]$ es mapeado al espacio RGB obteniendo el vector $\phi_p^* = [r_p^*, g_p^*, b_p^*]$.

En donde δ_s y δ_v son los umbrales para saturación e intensidad, respectivamente. Dada la naturaleza difusa del color, no hay valores específicos para decidir exactamente cuando un color es blanco o negro; de forma experimental encontramos que los mejores umbrales son $\delta_s = \mu_s - \sigma_s$ y $\delta_v = \mu_v - \sigma_v$; en donde μ_s y μ_v son la media de saturación e intensidad de la imagen, respectivamente; σ_s y σ_v son la desviación estándar de la saturación e intensidad de la imagen, respectivamente.

4. Experimentos y discusión

Recientemente la base de segmentación de Berkeley[1] (BSB) se está convirtiendo en la referencia para probar algoritmos de segmentación de imágenes de color [16]. Para los experimentos, implementados en Matlab 2014a, se seleccionó aleatoriamente un conjunto de 9 imágenes de las 300 imágenes que contiene la BSB, ver Fig. 4.

En la Fig. 5 se muestran las imágenes obtenidas al procesar las imágenes de la Fig. 4 empleando las RNCs con los diferentes tamaños que se indican. A su vez en la Fig. 6 se muestran las imágenes resultantes al ser procesadas las imágenes de la Fig. 4 empleando los MAOs con los tamaños que se indican.

Se puede observar fácilmente de las imágenes resultantes que estas pueden ser segmentadas solamente utilizando información de la cromaticidad; pero la segmentación también depende de la cantidad de neuronas de las RNs. Esto es, entre más grande sea la RN, mayor la cantidad de colores son reconocidos; de hecho, se puede apreciar que con los MAO se reconocen más colores que con las RNC, ya que se pueden observar más secciones o dentro de la imagen empleando los MAO. Aunque también, por lo mismo, hay algunas partes de la imagen que no son segmentadas homogéneamente.

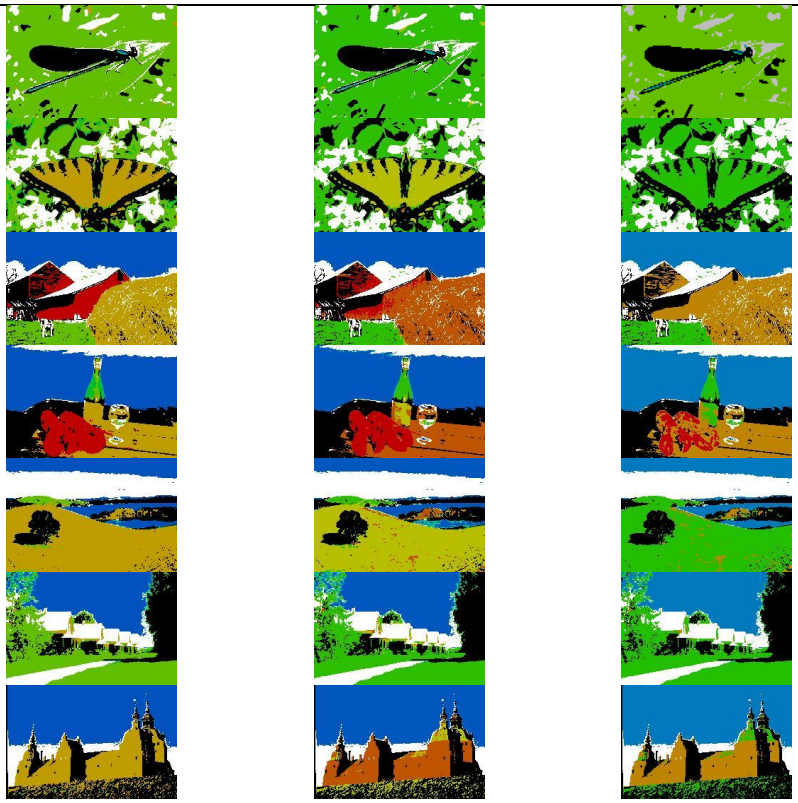
Al observar las imágenes se puede ver que estas tienen mejor segmentación empleando los MAOs. Por ejemplo, en las imágenes obtenidas al procesar la imagen 35070 con las RNCs, el fondo se combina con la hoja; mientras que la misma imagen procesada por los MAOs se puede ver que la hoja es segmentada del fondo, a pesar de tener ambos tonos verdes pero los MAOs son capaces de distinguir la diferencia de tonos.

Otro ejemplo, en las imágenes obtenidas de la imagen 35010 utilizando las RNCs, prácticamente el fondo de las hojas son segmentadas en verde, mientras que con los MAOs se pueden apreciar tonos en amarillo en los centros de las hojas. Las alas de la mariposa son segmentadas exitosamente con todas las redes neuronales, excepto utilizando la RNC de 25 neuronas, las cuales son segmentadas con el mismo tono de verde de las hojas del fondo. Una posible explicación es que esa red neuronal no "aprendió" correctamente a reconocer el tono amarillo durante el entrenamiento, por lo que la red neuronal debe ser entrenada nuevamente.

[1] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

Fig. 4. Imágenes extraídas de la BSB, empleadas para los experimentos realizados

RNC 9	RNC 16	RNC 25



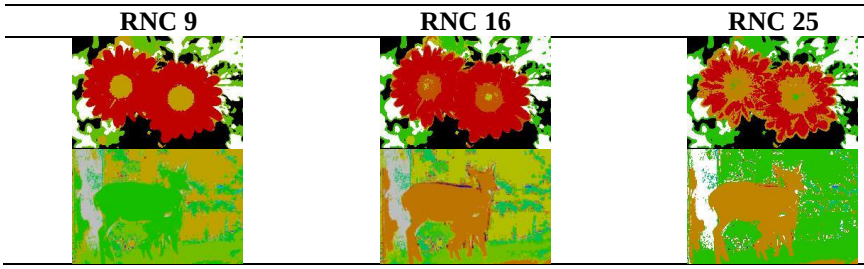
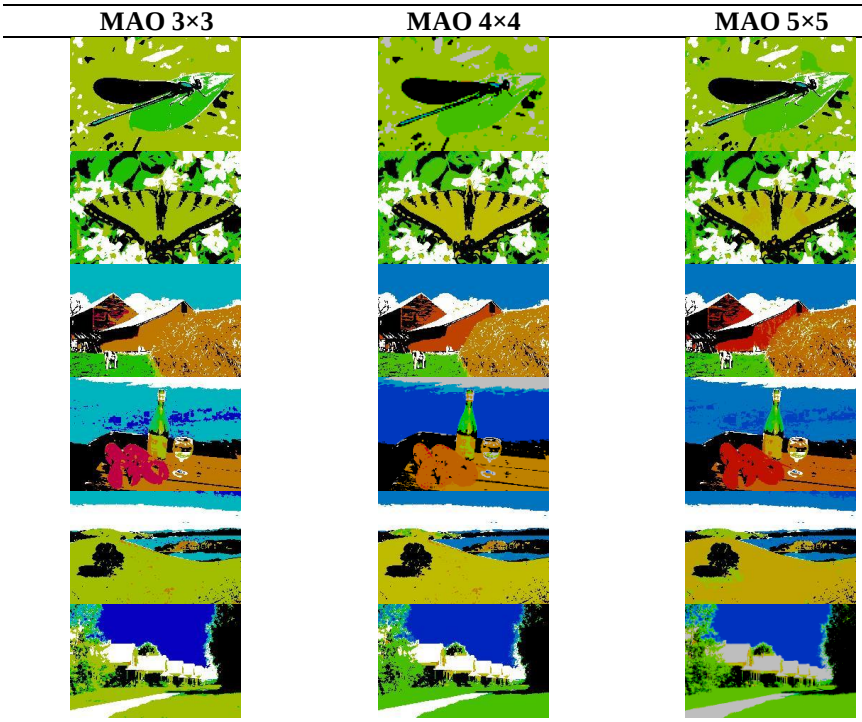


Fig. 5. Imágenes obtenidas empleando redes neuronales competitivas de 9, 16 y 25 neuronas

Dada la apariencia de las imágenes, la segmentación de las imágenes utilizando las RNCs es mejor cuando la red neuronal es pequeña, mientras que con los MAOs es lo contrario; es decir, el resultado de la segmentación de las imágenes utilizando los MAOs es mejor cuando la red neuronal es grande. Por ejemplo, la imagen 124084 obtenida con la RNC de 9 neuronas es muy parecida a la obtenida utilizando el MAO de 5×5 neuronas. Ocurre de forma similar con las imágenes 97010, 157032, 201080 y 35010 si se utiliza un la RNC de 9 neuronas y el MAO de 5×5 neuronas.

5. Conclusiones y trabajo futuro



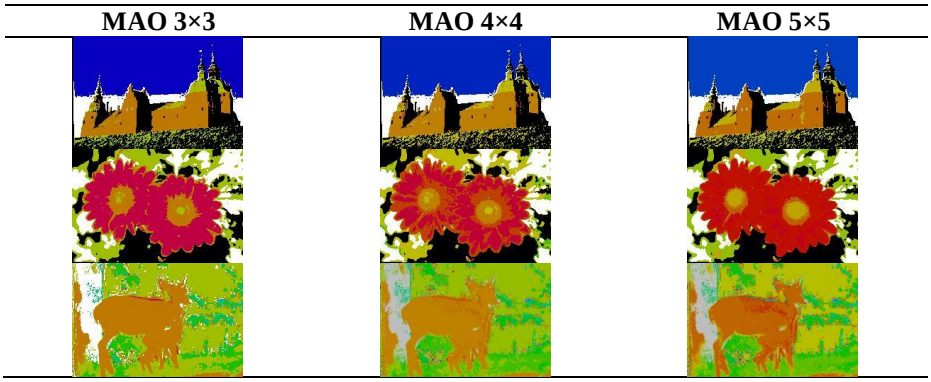


Fig. 6. Imágenes obtenidas empleando mapas auto-organizados de 3×3, 4×4 y 5×5 neuronas

En este trabajo se ha presentado una propuesta para segmentar imágenes por características de color. Se presentan pruebas empleando redes neuronales competitivas y mapas auto-organizados de diferentes tamaños, que son entrenadas con muestras de cromaticidad de diferentes colores; posteriormente se procesan las imágenes extrayendo sólo la cromaticidad de los colores de las imágenes, al mapear previamente las imágenes al espacio HSV. Cada pixel es agrupado con el tono de la neurona ganadora de la red neuronal, finalmente la imagen resultante es mapeada al espacio RGB.

Los mapas auto-organizados mostraron, en cuanto a la apariencia de la imagen, tener mejor desempeño que las redes neuronales competitivas; es decir, aunque la cantidad y forma de las secciones obtenidas empleando ambas redes neuronales son parecidas, los tonos asignados por los mapas auto-organizados se asemejan más a los de las imágenes originales.

La cantidad de colores que pueden reconocer las redes neuronales depende de la cantidad de neuronas que tienen. Las redes neuronales con pocas neuronas funcionan mejor en imágenes que tienen pocos colores; mientras que las redes neuronales con un número grande de colores tienen mejor desempeño con las imágenes que tienen varios colores.

Como trabajo futuro se contempla hacer una evaluación cuantitativa de la segmentación de las imágenes obtenidas con nuestra propuesta. Empleando las métricas de índice aleatorio probabilístico y de variación de información, que se están volviendo últimamente en las métricas estándar para medir el desempeño de los algoritmos de segmentación de color [16]. La imagen segmentada con nuestra propuesta es comparada con las imágenes segmentadas a mano que se encuentran en la BSB que sirven como referencia. Cada imagen de la BSB tiene un conjunto de 5 imágenes segmentadas a mano, con las que se hace la comparación. Por ejemplo, en la Fig. 7 se muestra las imágenes segmentadas a mano de la imagen 35010 de la BSB.

Fig. 7. Ejemplo de imágenes segmentadas a mano de la BSB

Las imágenes segmentadas obtenidas con nuestra propuesta son comparadas con cada una de las imágenes segmentadas a mano de la BSB, en donde la similitud de la segmentación se mide con las métricas mencionadas anteriormente. En la Fig. 8 se muestran dos ejemplos de imágenes segmentadas a mano, (a) y (c), y dos que se obtienen con nuestra propuesta, (b) y (d).

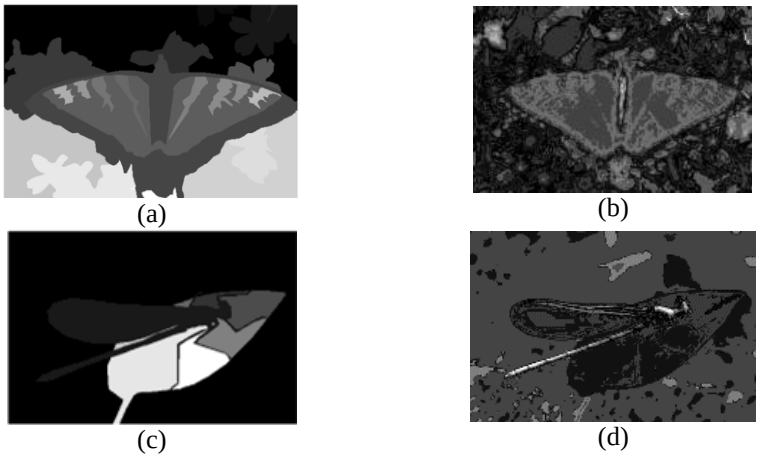


Fig. 8. Ejemplo de comparación de las imágenes: (a) y (c), segmentadas a mano y las segmentadas con nuestra propuesta, (b) y (d)

Por otra parte se contempla hacer pruebas con imágenes con ruido para observar el comportamiento de nuestra propuesta.

Agradecimientos. El primer autor agradece al CONACyT la beca para realizar estudios de maestría, con el número de registro 634201.

Referencias

1. Gökmen, V., Sügüt, I.: A non-computer vision based analysis of color in foods. Int. J. Foog Eng., Vol. 3, No. 5 (2007)
2. Lopez, J.J., Cobos, M., Aguilera, E.: Computer-based detection and classification of flaws in citrus fruits. Neural Comput. Appl., Vol. 20, No. 7, pp. 975–981 (2011)
3. Lespitö, L., Kuntuu, I., Visa, A.: Rock image classification using color features in Gabor space. J. Electron. Imaging, Vol. 14, No. 4, pp. 1–3 (2005)
4. Ghoeneim, D.M.: Optimizing automated characterization of liver fibrosis histological images by investigating color spaces at different resolutions. Theor. Biol. Med. Model., Vol. 8, No. 25 (2011)
5. Harrabi, R., Braiek, E.B.: Color image segmentation using multi-level thresholding approach and data fusion techniques: application in the breast cancer cells images. EURASIP J. Image Video Process., Vol. 11 (2012)
6. Wang, G., Man, L., Wang, B., Xiao, Y., Pan, W., Lu, X.: Fuzzy-based algorithm for color recognition of license plates. Pattern Recognit. Lett., Vol. 29, No. 7, pp. 1007–1020 (2008)
7. del Fresno, M., Macchi, A., Marti, Z., Dick, A., Clausse, A.: Application of color image segmentation to estrus detection. J. Vis., Vol. 9, No. 2, pp. 171–178 (2006)

8. Rotaru, C., Graf, T., Zhang, J.: Color image segmentation in HSI space for automotive applications. J. Real-Time Image Process., Vol. 3, No. 4, pp. 311–322 (2008)
9. Bianconi, F., Fernandez, A., Gonzalez, E., Saetta, S.A.: Performance analysis of color descriptors for parquet sorting. Expert Syst. Appl., Vol. 40, No. 5, pp. 1636–1644 (2013)
10. Aghbarii, Z.A., Haj, R.A.: Hill-manipulation: an effective algorithm for color image segmentation. Image Vis. Comput., Vol. 24, No. 8, pp. 894–903 (2006)
11. Mignotte, M.: A non-stationary MRF model for image segmentation from a soft boundary map. Pattern Anal. Appl., Vol. 17, No. 1, pp. 129–139 (2014)
12. Mousavi, B.S., Soleymani, F., Razmjooy, N.: Color image segmentation using neuro-fuzzy system in a novel optimized color space. Neural Comput. Appl., Vol. 23, No. 5, pp. 1513–1520 (2013)
13. Ong, S., Yeo, N., Lee, K., Venkatesh, Y., Cao, D.: Segmentation of color images using a two-stage self-organizing network. Image Vis. Comput., Vol. 20, No. 4, pp. 279–289 (2002)
14. Jiang, Y., Zhou, Z.H.: SOM ensemble-based image segmentation. Neural Process. Lett., Vol. 20, No. 3, pp. 171–178 (2004)
15. Wang, L., Dong, M.: Multi-level low-rank approximation-based spectral clustering for image segmentation. Pattern Recognit. Lett., Vol. 33, No. 16, pp. 2206–2215 (2012)
16. Mújica-Vargas, D., Gallegos-Funes, F.J., Rosales-Silva, A.J.: A fuzzy clustering algorithm with spatial robust estimation constraint for noisy color image segmentation. Pattern Recognit. Lett., Vol. 34, No. 4, pp. 400–413 (2013)
17. Huang, R., Sang, N., Luo, D., Tang, Q.: Image segmentation via coherent clustering in L*a*b* color space. Pattern Recognit. Lett., Vol. 32, No. 7, pp. 891–902 (2011)
18. Nadernejad, E., Sharifzadeh, S.: A new method for image segmentation based on fuzzy c-means algorithm on pixonal images formed by bilateral filtering. Signal Image Video Process., Vol. 7, No. 5, pp. 855–863 (2013)
19. Guo, Y., Sengur, A.: A novel color image segmentation approach based on neutrosophic set and modified fuzzy c-means. Circuits Syst. Signal Process., Vol. 32, No. 4, pp. 1699–1723 (2013)
20. Kim, J.Y.: Segmentation of lip region in color images by fuzzy clustering. Int. J. Control Autom. Sys., Vol. 12, No. 3, pp. 652–661 (2014)
21. Ito, S., Yoshioka, M., Omatu, S., Kita, K., Kugo, K.: An image segmentation method using histograms and the human characteristics of HSI color space for a scene image. Artif. Life and Robot., Vol. 10, No. 1, pp. 6–10 (2006)
22. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. 2nd ed. Prentice Hall (2002)

Red de transición aumentada y lenguaje formal para la danza Bhāratanāṭyam

Rosario Romero-Conde[1], Miguel Murguía-Romero[1,2]

[1] Grupo cultural *Nāṭya Sūtra*, Ciudad de México, México

[2] UNAM, FES Iztacala, México

miguelmurguia@ciencias.unam.mx

Resumen. El Bhāratanāṭyam es una danza clásica de la India, ancestral, se practica en muchos países, está documentada en textos antiguos y es muy elaborada: hay cerca de un centenar de *aḍavus* (movimientos básicos que en su conjunto componen una danza), 10 *maṇḍalas* (posiciones estáticas), y 10 grupos de *hastas* (gestos con las manos); también existe otro tipo de componentes, como las posiciones de los pies o movimientos de los ojos, entre otros. Así, esta danza es compleja porque incorpora múltiples aspectos corporales que deben ser dominados por el practicante cuando ofrece un recital. La definición de un lenguaje formal facilitará su documentación de manera sistemática, mejorando la comunicación entre sus practicantes y apoyando en su enseñanza. El objetivo de este trabajo fue definir un lenguaje formal básico para la danza Bhāratanāṭyam que pudiera ser procesado automáticamente por un analizador sintáctico, con la finalidad de facilitar el proceso de su enseñanza-aprendizaje. De todos los elementos de la danza, se seleccionó solo un subconjunto reducido para definir un léxico básico. Se construyó una red de transición aumentada (ATN) para definir el lenguaje formal que incluye nueve clases de símbolos terminales (categorías de palabras) y siete clases de símbolos no terminales o sub-redes. El diseño de lenguajes formales para danzas y su correspondiente gramática ATN, es una herramienta didáctica que auxilia en el proceso de enseñanza-aprendizaje, permitiendo también la documentación de danzas completas que pueden ser compartidas entre los practicantes.

Palabras clave: ATN, Danza Bhāratanāṭyam, Hinduismo, Procesamiento de Lenguaje Natural, Lenguaje de danza *Nāṭya Sūtra*, Gramáticas transformacionales.

Augmented Transition Network and Formal Language for Bhāratanāṭyam Dance

Abstract. Bhāratanāṭyam is an ancestral classical Indian dance whose practice has nowadays spread to many countries. The dance is documen-

ted in ancient texts and is very elaborated: there are around 12 types of items of dances, such as *alarippus*, *padams*, *jatisvarams*, or *tilanas*, more than a hundred of *aḍavus* (basic movements that in conjunction compound a dance), 10 *maṇḍalas* (static positions), and 10 types of *hastas* (hand gestures); there are also other components as feet positions, and eyes movements, among others. Thus, this ancestral dance is complex because it incorporates many corporal aspects that should be dominated by the practitioner in order to perform a recital. The definition of a formal language will facilitate the documentation of this dance in a systematic way that will help to improve the teaching-learning process and also the development and communication among the community of its performers. The objective of this work was to define a basic formal language for Bhāratanāṭyam that could be processed automatically by a syntactic analyzer, in order to facilitate the process of learning and teaching this ancestral Indian dance. From all elements of the dance, a basic subset was selected to define a base lexicon. An augmented transition network (ATN) was built to defined a formal language that includes nine classes of terminal symbols (words categories) and seven classes of non-terminal symbols or sub-networks. The design of formal languages for ancient dances, and its correspondent grammar through ATN, is a didactical tool that facilitates the teaching-learning process, and also the documentation of complete dances that can be shared among the practitioners.

Keywords: ATN, Bhāratanāṭyam dance, Hinduism, Natural Language Processing, *Nāṭya Sūtra* dance language, transformational grammars.

1. Introducción

El Bhāratanāṭyam es una danza ancestral, se le reconoce como una de las ocho danzas clásicas de la India y hoy en día se practica en muchos países [2]. Está documentada en textos ancestrales como el *Nāṭyaśāstra* de Bharata-Muni así como en los relieves en piedra de los templos hinduistas en donde se despliegan frases completas de movimientos. El Bhāratanāṭyam es una danza muy elaborada: existen cerca de 12 tipos de ítems de danzas como *alarippus*, *padams*, *jatisvarams*, o *tilanas*, más de cien *aḍavus* (movimientos básicos que en su conjunto conforman una danza), 10 *maṇḍalas* (posiciones estáticas), y 10 tipos de *hastas* (gestos de la mano; Tabla 1); también hay otro tipo de componentes como posiciones de los pies, movimientos de los ojos y del cuello, entre otros [9]. Así, esta danza ancestral es compleja porque incorpora muchos aspectos corporales que el practicante debe dominar al hacer un recital.

De manera típica la danza se enseña mediante ejemplo del profesor, el estudiante memoriza los ejercicios y toma notas con símbolos que inventa, la mayoría de las veces con dibujos caricaturizados del cuerpo humano. Este método tiene la ventaja de que permite al estudiante tomar notas de manera libre, sin embargo, es ineficiente cuando se desea representar danzas de duración larga que se componen

de varios movimientos. Además, este tipo de notación puede causar confusiones y malinterpretaciones.

El problema de construir un lenguaje para la danza ya se ha plantado con anterioridad. En los inicios del siglo pasado se propuso una notación para danza clásica [10], desafortunadamente el sistema es muy críptico por lo que no ha sido ampliamente usado. En México se propuso otro sistema para danza folklórica [12,11], pero la notación está orientada a representar la coreografía, más que los movimientos del bailarín.

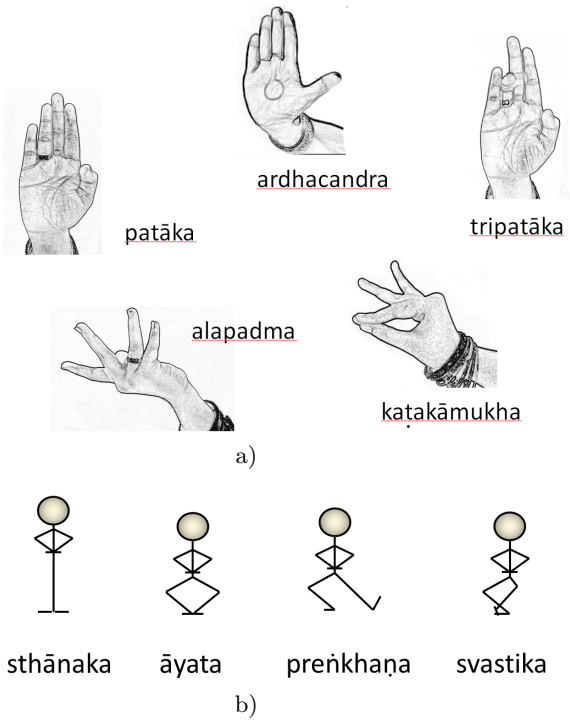


Fig. 1. Algunos ejemplos iconográficos de las posturas de la danza Bhāratanāṭyam. a) *hastas* (gestos de las manos); b) *maṇḍalas* (posiciones de las piernas)

1.1. Trabajos previos

Existen trabajos que aplican la informática al análisis de los movimientos y posturas del Bhāratanāṭyam, uno de ellos [8] propone codificar los pasos de la danza mediante un "vector de posición de danza" que consiste en una lista de números naturales que codifican treinta atributos que corresponden a sies partes del cuerpo, este vector de posición es más bien útil como una

representación interna más que un lenguaje para el practicante. Otro trabajo [7] que aplica la informática al análisis del Bhāratanāṭyam describe un sistema que reconoce gestos de las manos (*hastas*) mediante análisis de imágenes, y también la representación se orienta a la lectura por una computadora y no por el practicante.

La definición de un lenguaje formal facilitará la documentación de esta danza de manera sistemática y ayudará a mejorar el proceso de enseñanza-aprendizaje y también al desarollo y comunicación entre la comunidad de bailarines. Si el lenguaje formal se puede vincular con herramientas automáticas, entonces la documentación se podrá convertir en formas dinámicas suceptibles de ser analizadas y estudiadas en dispositivos electrónicos.

1.2. Objetivo

El objetivo de este trabajo fue definir un lenguaje formal para el Bhāratanāṭyam que pueda ser procesado automáticamente por un analizador sintáctico, con la finalidad de facilitar la documentación y el proceso de aprendizaje y enseñanza de esta danza ancestral de la India.

2. Danza y procesamiento de lenguaje natural

2.1. Redes de transición aumentada (ATN)

Una red de transción aumentada (ATN) es un viejo formalismo para el procesamiento del lenguaje natural (NLP) para analizar la estructura sintáctica de una oración, y fue construido con base en los trabajos seminales de las gramáticas generativas [4]. El formalismo [15,14] representa a una gramática por medio de gráficas compuestas por arcos y nodos, donde los nodos indican cada estado del análisis y los arcos imponen restricciones a la oración que es representada mediante una lista de palabras y símbolos de puntuación. Los arcos se etiquetan con símbolos terminales, i.e. palabras, o con nombres de sub-redes. Por ejemplo, si una oración está formada por una frase nominal seguida de una frase verbal, entonces la gráfica `SENTENCE` puede ser representada por los arcos: `NOUN_PHRASE` y `VERBAL_PHRASE`; a su vez, la gráfica `NOUN_PHRASE` puede estar conformada por los arcos `article` y `name`.

2.2. El uso de NLP para representación de danzas

El formalismo ATN deriva directamente de las redes de transición recursivas (RTN), ambas comparten la misma representación, la diferencia principal es que en las ATN se pueden especificar los rasgos y su concordancia, por ejemplo, la concordancia de género, de número y persona, entre otras. Esta característica del ATN puede ser explotada en un lenguaje para la danza porque los movimientos pueden ser exclusivos de partes específicas del cuerpo, i.e., puede existir concordancia entre tipos de movimientos y partes del cuerpo. Otra característica de las danzas, representadas como frases que cumplen restricciones, son los desplazamientos simétricos.

3. Método

La danza Bhāratanāṭyam tiene muchos elementos que se clasifican y describen sistemáticamente de forma tradicional, por ejemplo, los accesorios, la vestimenta, el maquillaje, o los movimientos del cuello y de los ojos, entre otros, que son considerados parte del lenguaje de la danza [13]. De todos los elementos de la danza, solo se eligió un subconjunto esencial para definir un léxico básico: diez *maṇḍalas*, dos tipos de *aḍavus* (*taṭṭa aḍavus*, y *naṭṭa aḍavus*, 16 en total), un tipo de *hastas* (28 en total; ver Tabla 1), siete direcciones de movimientos del cuerpo (`up`, `down`, `front`, `back`, `right`, `left`, y `sideway`), y cuatro posiciones de las piernas (Figura 1).

3.1. Rasgos y concordancia

Como muchos de los movimientos se refieren exclusivamente a los pies o a las manos, se incluyó a `hand/foot` como un rasgo de la red que impone condiciones de concordancia. Por ejemplo, en el léxico se describe para cada palabra, además de la categoría a la que pertence, el rasgo, mediante uno de los tokens `hand` o `foot`.

No obstante que la simetría de pies o manos es una característica importante de las secuencias de los movimientos, y por lo tanto es una buena candidata de rasgo, en esta primer versión de la ATN se decidió no incluirla.

3.2. Aḍavus como guía para definir el léxico

La descripción de los movimientos de piernas y manos se encuentra de forma muy detallada en los textos *Abhinayadarpana* de Nandikervara [6] y *Nāṭyaśāstra* de Bharata-Muni [3], en particular, los *aḍavus* son combinaciones de posiciones de piernas, formas de pararse, movimientos y *hastas* (gestos de las manos), y son las secuencias básicas de las que se compone una danza. Así, el léxico básico se creó tomando en cuenta las posiciones y movimientos de dos tipos de *aḍavus*: *taṭṭa aḍavus* que son pasos básicos que involucran movimientos sólo con las piernas, y los *naṭṭa aḍavus* que además involucran movimientos de brazos, incluyendo gestos con las manos.

3.3. Representación en lenguaje Prolog del analizador ATN

El léxico básico se agrupó en nueve categorías de palabras (Tabla 2) que se seleccionaron con base en dos criterios: 1) las palabras con las que se podrían describir los movimientos y partes del cuerpo involucrados en los *taṭṭa aḍavus* y *naṭṭa aḍavus*, y 2) las palabras que representan la lista de posiciones y movimientos básicos: *hastas* o gestos con las manos, y *maṇḍalas* o posiciones de los pies (Figura 1).

El predicado Prolog para representar las palabras es `word/3`:

```
word(Word_category, Word, Feature = <foot/hand/_>).
```

donde `Word_category` es la categoría de `Word`, v.gr. una preposición, es decir, de la categoría `preposition` (`at, to`) o `action_word` (`hit, rotate`) (Tabla 2); y `Feature` indica si la palabra `Word` tiene restricción de concordancia (`hand/foot`) o no (`_`). Sólo se consideró un rasgo para concordancia, que verifica que el movimiento sea congruente con la parte del cuerpo, por ejemplo, la posición `ardhacandra` es una *hasta*, i.e., un gesto de mano, entonces, la expresión `right hand ardhacandra` es válida, mientras `right leg ardhacandra` es inválida; los hechos Prolog usados para representar a las palabras involucradas son:

```
word(body_side, right, _).
word(body_part_word, hand, hand).
word(position, ardhacandra, hand).
word(body_part_word, leg, foot).
```

Todas las *asaṃyuta hastas* (Tabla 1) tienen asociada la etiqueta `hand`, y se representan con hechos del siguiente tipo `word(position, ardhacandra, hand)`. Algunos ejemplos de palabras que no tienen concordancia son:

```
word(preposition, at, _).
word(preposition, to, _).
word(body_part_word, head, _).
word(body_part_word, trunk, _).
```

Construimos el código del analizador ATN en lenguaje Prolog tomando como base el código del libro de texto de procesamiento de lenguaje natural de Gazdar y Mellish [5]. En el lenguaje Prolog los hechos se pueden representar directamente, y la máquina de backtrack automático facilita la programación del formalismo ATN, que básicamente consiste en la búsqueda de un camino en las redes, y que representa la solución del análisis sintáctico. Las redes se representan con el predicado `arc/4`:

```
arc(Node_from, Node_to, Word_or_Network, Network_of_this_arc)
```

donde `Node_from` y `Node_to` son el inicio y fin del arco `arc`, respectivamente, `Word_or_Network` es una palabra o la etiqueta de una red (`Network`), y `Network_of_this_arc` es el nombre de la red a la que pertenece el arco `arc`.

4. Resultados

4.1. El lenguaje formal propuesto: léxico y notación

El lenguaje formal incluye nueve clases de símbolos terminales (categorías de palabras; Tabla 2). Cada oración se forma por una secuencia de palabras, por ejemplo, las palabras de la oración:

```
stance mandala sthanaka; arms back hasta ardhacadra look at front;
end_stance.
```

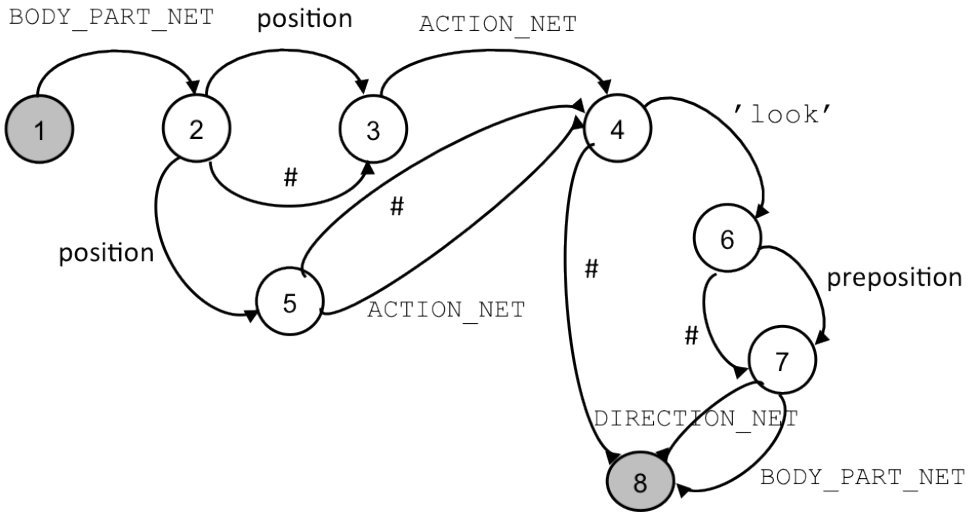


Fig. 2. La red BODY_PART_MOVEMENT_NET del analizador ATN. Analiza oraciones del Bhāratanāṭyam que involucran movimientos de piernas, manos, tronco, entre otros. Los nodos 1 y 8 son el inicial y el final de la red, respectivamente. El símbolo # denota un salto, i.e., el análisis continúa en el siguiente nodo del arco correspondiente

tienen las siguientes categorías:
`stance`.- constante que identifica una posición estática
`mandala`.- constante que identifica posiciones de las piernas
`sthanaka.- position`
`arms.- body_part_word`
`back.- direction_word`
`hasta.- body_part_word`
`ardhacadra.- position`
`look`.- constante que identifica que lo que sigue es un *dṛṣti bheda* (movimientos de los ojos)
`at.- prepositon`
`front.- direction`
`end_stance`.- constante que identifica el fin de la oración

Esta oración se puede entender mejor si se observa la Figura 1, y corresponde a los primeros elementos de la Figura 3a y 3b. La estructura sintáctica de la oración se incluye en la Figura 4.

4.2. La gramática ATN propuesta para la danza Bhāratanāṭyam

La gramática del lenguaje *Nāṭya Sūtra* contiene siete clases de símbolos no terminales, así, la ATN se compone de siete sub-redes (Tabla 3), que son:

PARTIAL_PHRASE_NET, BODY_PART_MOVEMENT_NET, STANCE_NET, BODY_PART_NET, ACTION_NET, DIRECTION_NET, y HIT_SEQUENCE_NET (Figura 2).

La red que procesa a las frases de partes del cuerpo es BODY_PART_NET, que consiste de tres nodos y tres arcos:

```
arc(1, 2, body_side, body_part).
arc(1, 2, '#', body_part).
arc(2, 3, body_part_word, body_part).
```

la red comienza en el nodo 1, y termina en el nodo 3:

```
begin(1, body_part).
end(3, body_part).
```

las dos plabaras body_side registradas en el léxico son:

```
word(body_side, right, _).
word(body_side, left, _).
```

y las palabras body_part_word registradas en el léxico son:

```
word(body_part_word, body, _).
word(body_part_word, trunk, _).
word(body_part_word, leg, foot).
word(body_part_word, legs, foot).
word(body_part_word, foot, foot).
word(body_part_word, feet, foot).
word(body_part_word, hand, hand).
word(body_part_word, hasta, hand).
word(body_part_word, hands, hand).
word(body_part_word, arm, hand).
word(body_part_word, arms, hand).
word(body_part_word, head, _).
word(body_part_word, neck, _).
word(body_part_word, mandala, foot).
```

En la Figura 3 se muestra el *taṭṭa aḍavu* 5 en el lenguaje *Nāṭya Sūtra* y la estructura sintáctica de sus frases. Como un ejemplo, para la siguiente frase:

```
stance mandala sthanaka; arms back hasta ardhacandra look at front;
end_stance.
```

la salida del procesamiento por la ATN, que es la estructura sintáctica propuesta, se muestra en la primer columna de la Figura 4.

5. Discusión

El lenguaje *Nāṭya Sūtra* para Bhāratanāṭyam es una herramienta para la enseñanza y el aprendizaje de esta danza ancestral, además, mediante este lenguaje se pueden documentar danzas completas. La gramática propuesta incluye

un léxico base que puede ampliarse para incorporar más elementos como son movimientos del cuello y ojos, entre otros. El manejo de rasgos se ejemplifica con solo una característica, pero se podrán incorporar más rasgos, como por ejemplo la simetría: proponemos que una `COMPLETE_PHRASE` puede descomponerse en dos `PARTIAL_PHRASE` que se complementen una a otra en simetría, derecha e izquierda, y que una `DANCE` se forme de un conjunto de `PARTIAL_PHRASE`, y que contenga cuando menos una `COMPLETE_PHRASE`, es decir, que contenga al menos una secuencia derecha que se repita en su análoga izquierda.

Con el lenguaje formal propuesto se pueden representar danzas ancestrales y puede usarse para soportar las dos posiciones dialécticas: ortodoxia e innovación. Para los innovadores, los beneficios de usar gramáticas ATN para danzas ancestrales se pueden visualizar fácilmente, por ejemplo, se podrían crear nuevas secuencias, incluso danzas completas que siguieran las reglas impuestas en la gramática; por otra parte, para los ortodoxos, nos gustaría considerar el punto de vista de la gran bailarina Balasaraswati quien dijo que "las más grandes autoridades de la danza han reconocido definitivamente que es la ortodoxia de la disciplina tradicional la que da la más amplia libertad para la creatividad individual del bailarín" [1]. Nosotros comulgamos con esa idea, y proponemos al formalismo ATN y al lenguaje *Nāṭya Sūtra* como una herramienta para documentar y preservar al Bhāratanāṭyam.

6. Trabajo futuro

Hay diversos elementos de la danza aún no incorporados en el lenguaje que requieren su adaptación y nomenclatura, y quizá la definición de nuevas categorías de palabras y subredes gramaticales. La capacidad de los ATN para procesar rasgos y que en esta primera versión del lenguaje se ejemplifica con la verificación de concordancia entre manos y pies, puede ser mejor aprovechada incorporando restricciones adicionales, como por ejemplo la simetría izquiera/derecha en la secuencia de los movimientos, característica de esta danza.

7. Conclusiones

El diseño de lenguajes formales para danzas ancestrales y sus correspondientes gramáticas mediante redes de transición aumentada, constituyen herramientas que pueden facilitar el proceso de enseñanza-aprendizaje, así como la documentación de danzas completas que pueden ser compartidas entre sus practicantes.

El lenguaje *Nāṭya Sūtra* descrito en este trabajo puede ser de utilidad a los practicantes de la danza Bhāratanāṭyam, tanto maestros como alumnos. Los nombres cortos para las *asaṃyuta hastas*, o gestos de una mano, y las categorías de palabras del lenguaje propuesto constituyen herramientas útiles al tomar notas ante una presentación, permitiendo el registro más rápido que de la manera tradicional mediante caricaturas del cuerpo humano.

Agradecimientos Nuestro querido *upādhyāyaḥ*, el Dr. Roberto García Fernández de Sánscrito en México, revisó la transliteración de las palabras en lengua sánscrita. Gabriela Murguía-Romero revisó el texto de una versión preliminar del manuscrito. RRC y MMR agradecen a Patricia Torres (ICCR Gurudev Tagore México, Gobierno de la India), Fabiola Ocón (Danzas de la India) y Julieta Solís (Opus Uno), y MMR a Sagrario T. Gómez y Natalia Cárdenas (Dirección General de Danza, UNAM), por sus enseñanzas sobre Bhāratanāṭyam.

Referencias

1. Balasaraswati, T.: On Bharata Natyam. Dance Chronicle 2, 106–116 (1978)
2. Balasubramaniam, C.: Language of the soul. Published by Public Diplomacy Division, New Delhi, for the Ministry of Externals Affairs, India. November-December 26, 58–63 (2012)
3. Bharata-Muni: The Nāṭyaśāstra. Transalted into English by Manmohan Ghosh. Bibliotecha Indica. Iissue no. 1559, work no.272. Asiatic Society of Bengal. Calcuta (1951)
4. Chomsky, N.: On certain formal properties of grammars. Information and Control 2, 137–167 (1959)
5. Gazdar, G., Mellish, Ch.S.: Natural language processing in Prolog: an introduction to computational linguistics. Addison-Wesley (1989)
6. Ghosh, M.E.: Nandikesvaras Abhinayadarpanam: a manual of gesture and posture used in Hindu dance and drama. KL Mukhopadhyay. Calcuta (1957)
7. Hariharan, D., Acharya, T., Mitra, S.: Recognizing hand gestures of a dancer. In: Kuznetsov, S. O. (ed), PReMI 2011, LNCS 6744. Springer-Verlag Berlin Heidelberg. pp. 186–192 (2011)
8. Jadhav, S., Joshi, M., Pawarm, J.: Modeling BharataNatyam dance steps: Art to smart. In: Proceedings of the CUBE International Information Technology Conference 2012. pp. 320–325. Pune, Maharashtra, India (September 2012)
9. Krishna Rao, U., U.K. Chandrabhaga Devi : A Panorama of Indian Dances. Sri Satguru Publications (1993)
10. von Laban, R.: Coreografía. Primer cuaderno. Eugen Diederichs, Jena, Germany (1926). Translated by Carla Doniz Geist, Instiuto Nacional de Bellas Artes y Literatura, México. 141pp (2013)
11. Nuñez Mesta, M.A., Reyes Gómez, L., de Anda Esquivel, F.: Bailes del folklor mexicano, 5 vols. Trillas, México (2001)
12. Nuñez Mesta, M.A., Reyes Gómez, L., de Anda Esquivel, F.: Bailes del folklor mexicano: metodología de la enseñanza mediante el sistema ACADEDA. Trillas, México (2001)
13. Venkataram, L.: Harya in classical dances. India Perspectives. Published by Public Diplomacy Division, New Delhi, for the Ministry of Externals Affairs, India. November-December 24, 67–79 (2010)
14. Winograd, T.: Language as a Cognitive Process. Vol. 1. Syntax. Addison-Wesley (1983)
15. Woods, W.A.: Transition network grammars for natural language analysis. Commun. ACM 13, 591–606 (1970)

Tabla 1. *Asaṃyuta hastas* (gestos de una mano) usados en la danza Bhāratanāṭyam y sus correspondientes símbolos propuestos en el lenguaje formal *Nāṭya Sūtra*. Proponemos que los símbolos cortos se formen concatendando la letra 'H' a las cuatro primeras letras del nombre sánscrito, excepto para la hasta *haṃsapakṣa*, para diferenciarla de *haṃsasya*

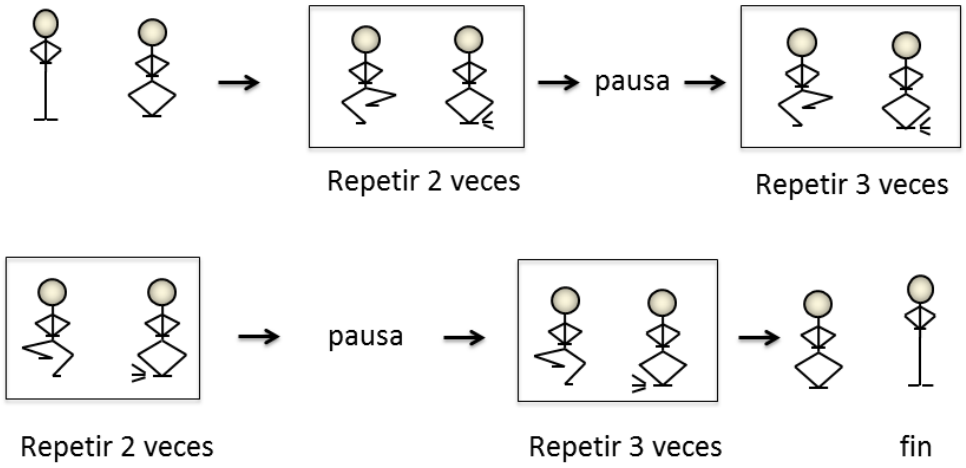
	Nombre en sánscrito	**Nombre corto**	**Símbolo de átomo Prolog**	**Símbolo corto Prolog**	**Símbolo Prolog alterno**
1	patāka	Hpata	pataka	hpata	h01
2	tripatāaka	Htrip	tripataka	htrip	h02
3	ardhapatāka	Hardh	ardhapataka	hardh	h03
4	kartarīmukha	Hkart	kartarimukha	hkart	h04
5	mayūra	Hmayu	mayura	hmayu	h05
6	ardhacandra	Hardh	ardhacandra	hardh	h06
7	arāla	Haral	arala	haral	h07
8	śukatuṇḍa	Hsuka	sukatunda	hsuka	h08
9	muṣṭi	Hmust	musti	hmust	h09
10	śikhara	Hsikh	sikhara	hsikh	h10
11	kapittha	Hkapi	kapittha	hkapi	h11
12	kaṭakāmukha	Hkata	katakamukha	hkata	h12
13	sūcī	Hsuci	suci	hsuci	h13
14	candrakalā	Hcand	candrakala	hcand	h14
15	padmakośa	Hpadm	padmakosa	hpadm	h15
16	sarpaśiras	Hsarp	sarpasiras	hsarp	h16
17	mṛgaśīrṣa	Hmrga	mrgasirsa	hmrga	h17
18	siṃhamukha	Hsimh	simhamukha	hsimh	h18
19	kaṅgula	Hkang	kangula	hkang	h19
20	alapadma	Halap	alapadma	halap	h20
21	catura	Hcatu	catura	hcatu	h21
22	bhramara	Hbhra	bhramara	hbhra	h22
23	haṃsasya	Hhams	hamsasya	hhams	h23
24	haṃsapakṣa	Hhamp	hamsapaksa	hhamp	h24
25	sandamśa	Hsand	sandamsa	hsand	h25
26	mukula	Hmuku	mukula	hmuku	h26
27	tāmracūḍa	Htamr	tamracuḍa	htamr	h27
28	triśūla	Htris	trisula	htris	h28

Tabla 2. Categorías de palabras definidas en el lenguaje formal *Nāṭya Sūtra*

Categoría de palabra	**Descripción** `[Ejemplos de palabras]`
`body_part_word`	La parte del cuerpo para la que se describe un movimiento `[foot]`, `[hand]`, `[head]`, `[leg]`, `[trunk]`, ...
`body_side`	Especifica si se refiere a la parte derecha o izquierda `[left]`, `[right]`, por ejemplo, usadas en las expresiones `[left hand]`, o `[right leg]`.
`action_word`	La acción que se realiza por alguna parte del cuerpo `[hit]`, `[rotate]`.
`preposition`	Preposición opcional que se puede usar antes de una dirección [`at, to`], por ejemplo, en expresiones como `[to left]`, `[at front]`, ...
`position`	Posiciones de las piernas (*maṇḍalas*) `[sthanaka]`, `[svastika]`, ..., posiciones estandarizadas de piernas, brazos y manos, y *hastas* `[pataka]`, `[ardhacandra]`, ...
`direction_word`	Direcciones básicas en el espacio `[down]`, `[up]`, `[left]`, `[right]`, ...
`adavu`	Secuencias básicas estandarizadas de movimientos `[tatta_adavu_1]`, `[tatta_adavu_2]`, ... `[natta_adavu_1]` ...
`n`	Números naturales `[1, 2, 3, 4, 5, 6, 7, 8, 9, ...]`
`separator`	Separador opcional de secuencias de movimientos `[;]`

Tabla 3. Redes definidas en el lenguaje formal *Nāṭya Sūtra*

Red	**Descripción {**`ejemplo de frase reconocida`**}**
`partial_phrase_net`	Frases o secuencias de una o más `body_part_movement` o `stance` (ver red `body_part_movement_net` y `stance_net`): `[stance mandala sthanaka arms back hasta ardhacandra look at front end_stance; stance mandala aayata arms back hasta ardhacandra end_stance.]`
`body_part_movement_net`	Realiza el "parsing" de las frases acerca del movimiento de una parte específica del cuerpo, como `feet`, `trunk`, `arms`, `hands`: `[mandala aayata right foot hit.]`
`stance_net`	Muy similar a `body_part_movement_net`, usada para describir posiciones no dinámicas del cuerpo, i.e., la posición de partes del cuerpo en un instante, y no en una secuencia a través del tiempo: `[stance mandala sthanaka arms back hasta ardhacandra look at front end_stance.]`
`body_part_net`	Identifica partes del cuerpo: `[right foot.]`, `[trunk.]`, `[legs.]`, `[left hand.]`, `[head.]`
`action_net`	Identifica acciones que principalmente se realizan con pies y manos: [`rotate.`], `[hit.]`
`direction_net`	Identifica las direcciones en el espacio de los movimientos: `[right.]`, `[down front.]`
`hit_sequence_net`	Identifica acciones de golpeteo con el pie: `[left hit 2 speed 1 pause hit 3 speed 2.]`



a)

```
% Preparacion
stance mandala sthanaka; arms back hasta ardhacandra look at front; end_stance.
stance mandala aayata; arms back; hasta ardhacandra; end_stance.
% tatta adavu 5
mandala aayata right hit 2 speed 1 pause hit 3 speed 2.
mandala aayata left hit 2 speed 1 pause hit 3 speed 2.
% Final
stance mandala sthanaka; arms back; hasta ardhacandra look to front; end_stance.
```

b)

```
?- atn([
% PREPARACION
stance, mandala, sthanaka, arms, back, hasta, ardhacandra, look, at, front, end_stance,
stance mandala, aayata, arms, back,  hasta, ardhacandra, end_stance,
% TATTA ADAVU 5
mandala, aayata, right, hit, 2, speed, 1, pause, hit, 3, speed, 2,
mandala, aayata, left, hit, 2, speed, 1, pause, hit, 3, speed, 2,
% FINAL
stance, mandala, sthanaka, arms, back, hasta, ardhacandra, look, to, front, end_stance
], Parse).
```

c)

Fig. 3. Ejemplo de una oración del lenguaje *Nāṭya Sūtra*. a) Secuencia iconográfica del *taṭṭa aḍavu* 5; b) La oración del *taṭṭa aḍavu* 5 escrita en lenguaje *Nāṭya Sūtra*; c) Representación Prolog de la oración en a) y b) aceptada como entrada del ATN

```
partial_phrase>
   stance>
      word_stance: stance
      body_part_movement>
         body_part>
            body_part_word: mandala
         position: sthanaka
      body_part_movement>
         body_part>
            body_part_word: arms
         action>
            direction>
               direction_word: back
      body_part_movement>
         body_part>
            body_part_word: hasta
         position: ardhacandra
         word_look: look
         action>
            preposition: at
            direction>
               direction_word: front
      word_end_stance: end_stance
   stance>
      word_stance: stance
      body_part_movement>
         body_part>
            body_part_word: mandala
         position: aayata
      body_part_movement>
         body_part>
            body_part_word: arms
         action>
            direction>
               direction_word: back
      body_part_movement>
         body_part>
            body_part_word: hasta
         position: ardhacandra
      word_end_stance: end_stance
   body_part_movement>
      body_part>
         body_part_word: mandala
      position: aayata
      action>
         hit_sequence>
            body_side: right
            word_hit: hit
            n: 2
            word_speed: speed
            n: 1
            word_pause: pause
            word_hit: hit
            n: 3
            word_speed: speed
            n: 2
   body_part_movement>
      body_part>
         body_part_word: mandala
      position: aayata
      action>
         hit_sequence>
            body_side: left
            word_hit: hit
            n: 2
            word_speed: speed
            n: 1
            word_pause: pause
            word_hit: hit
            n: 3
            word_speed: speed
            n: 2
   stance>
      word_stance: stance
      body_part_movement>
         body_part>
            body_part_word: mandala
         position: sthanaka
      body_part_movement>
         body_part>
            body_part_word: arms
         action>
            direction>
               direction_word: back
      body_part_movement>
         body_part>
            body_part_word: hasta
         position: ardhacandra
         word_look: look
         action>
            preposition: to
            direction>
               direction_word: front
      word_end_stance: end_stance
```

Fig. 4. Estructura sintáctica de la oración de la Figura 3 producida por el analizador ATN

Segmentación automática de billetes mexicanos basada en un modelo de color y referencias geométricas

Juan Pablo Flores-Mendoza, Alfonso Rojas-Domínguez,
Rafael López- Leyva, Manuel Ornelas-Rodríguez,
Raúl Santiago-Montero

Tecnológico Nacional de México - Instituto Tecnológico de León,
León, Gto.,
México

{juan.pbl.mdza, alfonso.rojas, leyvarafael24}@gmail.com,
mornelas67@yahoo.com.mx, raul.santiago@itleon.edu.mx

Resumen. Recientemente, el desarrollo de sistemas de ayuda para discapacitados visuales basados en procesamiento de imágenes y reconocimiento de objetos ha captado interés. En este trabajo se presenta un método para segmentación automática de billetes mexicanos basado en color y referencias geométricas (puntos de interés obtenidos automáticamente). Nuestro objetivo es identificar con precisión la región del billete para extraer información de color, útil para la posterior clasificación del billete. El método consiste en: 1) dibujar una región de referencia automáticamente; 2) generar un modelo de color y una imagen de similitud de color afectada por una función de peso geométrica; 3) umbralizar la imagen de similitud para obtener la segmentación final. El método fue evaluado sobre un conjunto de 926 imágenes de distintos billetes y tomadas bajo distintas condiciones de iluminación; en esta prueba el método demostró ser rápido y lograr un alto desempeño.

Palabras clave: Segmentación por color, segmentación automática, modelo de color estadístico, sistema de ayuda para débiles visuales.

Mexican Banknote Segmentation Based on a Color Model and Geometric References

Abstract. Recently, the development of aiding systems for visually impaired people, based on image processing and object recognition, has attracted interest. In this work, a method for automatic segmentation of Mexican banknotes based on color and geometric references (points of interest obtained automatically) is presented. Our objective is to precisely identify the banknote region in order to extract color information useful for the later classification of the banknote. The method consists of: 1) automatically draw a region of reference; 2) generate a color model and a similarity image affected by a geometric weight function; 3) thresholding of the similarity image to produce the final segmentation. The method was evaluated on a set of 926 images of banknotes obtained under

different illumination; the method showed to be fast and achieve high performance.

Keywords: Color based segmentation, automated segmentation, statistical color model, aiding system for the visually impaired.

1. Introducción

En todo el país (México) la segunda discapacidad más frecuente registrada es la discapacidad visual, después de las deficiencias motoras. En 2010 la población del país fue de 112 millones de personas, de las cuáles 1 millón de personas reportaron sufrir discapacidad visual [1]. Esta información nos ha motivado a desarrollar un sistema de ayuda para discapacitados visuales. Una de las tareas que es de gran importancia para la comunidad con discapacidad visual es el reconocimiento de la denominación de los billetes, ya que esta tarea es difícil llevarla a cabo basándose solo en información táctil. El desarrollo de un sistema de ayuda a débiles visuales capaz de realizar el reconocimiento automático de billetes es un proceso complejo que implica varias etapas de procesamiento. En este trabajo, nuestra atención se centra en la tarea de segmentación de billetes de banco mexicanos usando su información de color y en una técnica para la detección automática de puntos de interés que se ha descrito en [2].

Las técnicas de segmentación se utilizan a menudo en aplicaciones de visión artificial. El proceso de segmentación permite que la descomposición de una escena en un objeto en primer plano y un fondo para el análisis de la escena. Ejemplos de aplicaciones de visión artificial que hacen uso de técnicas de segmentación son: el diagnóstico médico, la localización de objetos, reconocimiento de huellas dactilares, reconocimiento de caras, etc. En este documento, la aplicación discutida es el desarrollo de un sistema de asistencia para débiles visuales que sea capaz de reconocer objetos de uso común tales como billetes. El método propuesto consiste, primeramente, en la obtención de un modelo estadístico de color que se genera a partir de una región semilla identificada de forma automatizada; luego, una medida de similitud calculada entre cada píxel de la imagen de un billete y el modelo de color permite definir la segmentación final.

El resto del trabajo se organiza de la siguiente manera: En la sección 2 se presenta una revisión del estado del arte sobre distintas metodologías de sistemas para ayuda a invidentes que incluyen reconocimiento de billetes. El método de segmentación propuesto se describe en la Sección 3. Los resultados experimentales se presentan en la Sección 4 y estos resultados se discuten en la Sección 5. Por último, la Sección 6 contiene nuestra conclusión y las direcciones de trabajo futuro.

2. Trabajos relacionados

Actualmente las aplicaciones centradas en el reconocimiento de los billetes en el mundo implementan diferentes técnicas de segmentación de imágenes en las que se usa color, textura, técnicas de umbralización, redes neuronales, etc. En [3], se describe la clasificación de billetes de México mediante la extracción de su color dominante en el

espacio RGB para máquinas expendedoras. Esto se complementa con Patrones Binarios Locales (una representación de textura). La desventaja de este método es que requiere de condiciones de iluminación controlada mientras que una aplicación para ayuda a invidentes implica manejar condiciones no controladas de iluminación. Dicho trabajo es el único que discute la clasificación de billetes mexicanos en la literatura.

Existen dispositivos de ayuda para reconocer la denominación de billetes. Por ejemplo, "*Currency Note Recognizer*" [4] es un dispositivo con un entorno controlado para clasificar billetes de Malasia con un 70%-90% de éxito. Sin embargo, dicho dispositivo se encuentra limitado a esta tarea específica. En cambio en este trabajo presentamos un módulo de un sistema con múltiples funciones (reconocimiento de objetos, navegación, etc) además del reconocimiento de billetes. La apariencia de los billetes mexicanos y un prototipo del sistema de ayuda se muestran en la Figura 1.

Otro ejemplo es "*Bionic Eyeglass*" [5], una aplicación diseñada para una plataforma móvil que incluye un módulo dedicado a realizar la clasificación de billetes el cual extrae una serie de características (morfológicas, de marcas geométricas, de color, de retratos de billetes y denominaciones, etc.). Este trabajo demuestra que pueden emplearse modelos estadísticos de color para discriminar objetos de colores conocidos. Por otro lado, la desventaja de este método, como lo mencionan sus autores, se encuentra en la alta complejidad computacional de los algoritmos, que obliga a utilizar un procesador externo al del dispositivo móvil. Además, interviene un gran número de parámetros que afectan el rendimiento y sus valores no fueron bien definidos.

Fig. 1. Izquierda: Aspecto de los billetes mexicanos. Derecha: Sistema de ayuda

Respecto a la segmentación basada en color de objetos además de billetes, existen varios ejemplos. Un algoritmo para la segmentación de imágenes basado en la similitud de color propuesto en [6] utiliza una medida llamada "*SIMILATION*" con la cual se etiquetan píxeles de una imagen para su segmentación. A pesar de que este trabajo enfatiza medir la similitud entre colores, encontramos que la tal medida no es lo suficientemente robusta como para hacer frente a condiciones variables de iluminación. Además, no toma en cuenta otras características de los objetos (como la geometría), que en el caso de los billetes son muy útiles para su segmentación.

Otro método que utiliza el color como una característica principal para segmentar imágenes se describe en [7]. En tal método el usuario marca un área para seleccionar los colores representativos del objeto a segmentar; posteriormente con un algoritmo de agrupamiento (*mean shift*) se fragmenta la imagen; se calcula la similitud de color entre los distintos fragmentos; a continuación se fusionan los fragmentos similares a los seleccionados previamente por el usuario y por separado se fusionan entre sí aquellos que son distintos o que se encuentran alejados de la selección del usuario. Aunque este método muestra buenos resultados, no es posible su utilización en nuestro sistema debido a que el tiempo de procesamiento es muy elevado para ser empleado en un sistema en tiempo real y también debido a que un usuario invidente no puede realizar la selección de los colores representativos de manera precisa.

En este trabajo se propone una técnica basada en color (en el espacio RGB) para la segmentación de los billetes mexicanos en imágenes digitales. Esta idea se basa en la combinación de color y referencias geométricas obtenidas de forma automática, puesto que nuestra revisión de la literatura indica que los mejores resultados se obtienen en función del color y otras técnicas complementarias. La siguiente sección describe los detalles de nuestra propuesta.

3. Metodología

En este trabajo se propone un método de segmentación de imágenes que tiene el objetivo de separar la región de un billete presente en una imagen del fondo de la escena. El método propuesto se basa en tres etapas: 1) la definición de una región semilla con base en los puntos de interés; 2) la generación de una imagen de similitud de color basada en un modelo derivado de la región semilla; 3) La aplicación de una función de peso y de un umbral sobre la imagen de similitud, para producir la Región De Interés (RDI) final. Cada una de estas etapas se describe a continuación.

3.1. Definiendo una región semilla

A lo largo de este trabajo se utiliza el término "región semilla" para referirse a una región en el espacio de la imagen de la cual se obtiene una muestra de información de color de los píxeles que se utilizará en etapas posteriores. La definición de la región semilla consta de dos pasos: el primer paso es la extracción de una muestra inicial de color; el segundo paso es la ampliación de la región inicial para robustecer la muestra de color. Claramente, esta idea requiere que la muestra de color inicial sea obtenida de los píxeles contenidos en el billete de la imagen. Garantizamos esto mediante la detección de rasgos característicos de un billete usando un método geométrico que hemos presentado anteriormente [2]. En este trabajo se resumen brevemente los puntos más importantes de tal procedimiento, ya que es un paso necesario en nuestra metodología; se remite al lector a [2] para una exposición completa del método empleado. Para ayudar al lector a seguir la exposición de nuestra metodología presentamos un esquema general de nuestra propuesta en la Fig. 2

Muchos billetes de todo el mundo utilizan los números indo-arábigos para representar su denominación [8]. Algunos autores (e.g. [5]) han presentado estrategias diseñadas para extraer esta información y utilizarla para la clasificación de billetes.

Nuestra propia estrategia consiste en detectar las regiones circulares o casi- circulares en los números impresos, en particular los de los ceros que están presentes en la mayoría de los valores de los billetes y en los numerales 2 y 5 que también se utilizan en la mayoría de las denominaciones monetarias en todo el mundo.

Para la detección de dichas regiones circulares hemos empleado la Transformada de Simetría Radial (FRS) desarrollado por Loy y Zelinsky [9]. La transformada FRS es un detector de puntos de interés que permite la detección de lugares con máxima simetría radial. Anteriormente se demostró su eficacia para la detección de números de billetes de banco [2].

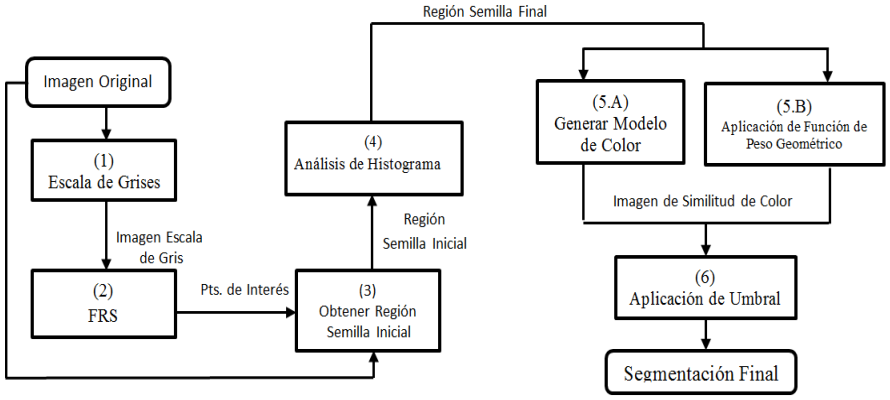


Fig. 2. Diagrama general del proceso de segmentación de imágenes

En este trabajo, con el fin de presentar y evaluar un método que es independiente de la exactitud de la etapa de detección, los puntos de interés correspondientes se calcularon y se validaron previamente (pasos 1 y 2 en la Fig.2). Es importante señalar que nuestro objetivo es la fusión de atributos de distintos dominios, tales como la geometría y el color de los billetes para darle robustez a la etapa de clasificación. Esta misma estrategia ha sido utilizada por otros autores (cf. [3, 5]). La detección de los puntos de interés por medio de la transformada FRS se ilustra en la Fig. 3; nótese que en las imágenes de prueba el usuario presenta sólo un billete de banco a la vez a la cámara de sistema. La detección de puntos de interés está regida por un parámetro que le indica a la transformada el tamaño aproximado de las regiones que debe detectar, de modo que se evitan detectar regiones espurias (ver detalles en [2]).

Basándose en los puntos de interés detectados es posible definir una región que se encuentra dentro del billete (paso 3 en la Fig.2). Esto se logra de la siguiente manera: en primer lugar se calcula el ángulo que se forma entre cada par de puntos de interés (es decir, sus coordenadas en pixeles) y se selecciona el par de puntos que definen el segmento más cercano a la horizontal. En segundo lugar, se calcula la magnitud M del segmento de línea entre los puntos seleccionados, de manera que un vector unitario $\boldsymbol{u} = [u_x, u_y]$ que apunta en la misma dirección de dicha línea pueda ser definido.

Por último, se forma una región rectangular mediante la asignación de uno de los puntos seleccionados como una de las esquinas ($\boldsymbol{C}_1 = [C_1^x, C_1^y]$) y usando las siguientes ecuaciones para definir las coordenadas de las tres esquinas restantes:

$$\begin{aligned} C_2^x &= C_1^x - h \cdot u_x M \\ C_2^y &= C_1^y - h \cdot u_y M \\ C_3^x &= C_1^x + v \cdot u_y M \\ C_3^y &= C_1^y - v \cdot u_x M \\ C_4^x &= C_2^x + v \cdot u_y M \\ C_4^y &= C_2^y - v \cdot u_x M \end{aligned} , \quad (1)$$

donde *v* y *h* son constantes que controlan la proporción de los lados verticales y horizontales del rectángulo, respectivamente, con base a la longitud de la línea entre los puntos de referencia, *M*. En este trabajo, nos basamos en la proporción entre los lados de los billetes mexicanos para establecer los valores de *v*= 10 y *h*=1. Estas proporciones garantizan que el rectángulo producido se encuentre dentro de los billetes; para otros billetes del mundo éstas deben ajustarse según las medidas correspondientes. La región rectangular producida de este modo nos ofrece una muestra de colores de los píxeles, I. Tras haber obtenido esta muestra de color inicial, definiremos una segunda región que corresponde a una muestra mejorada. La segunda región, conocida como Región Semilla, *S*, es una región rectangular extendida que ocupa una gran proporción del billete. La región semilla se obtiene de la siguiente manera.

Fig. 3. Detección de las regiones semi-circulares que corresponden a la denominación de los billetes a través de la transformada FRS. Figura principal: Imagen de prueba sobre la cual ha sido superpuesto los máximos de la transformada FRS. El contraste de la imagen se ha exagerado con fines ilustrativos. Recuadro: versión ampliada de la región de denominación de $ 200

Dado que la denominación de los billetes se encuentra en una esquina que puede ser en el lado izquierdo o el lado derecho del billete (dependiendo de si el usuario presenta a la cámara una cara del billete de banco o la otra), podemos definir dos regiones candidatas para convertirse en región semilla: si los puntos de referencia se encuentran en la esquina izquierda, entonces un rectángulo que se extiende a la derecha de esta esquina se convertiría en la región semilla; por el contrario, si los puntos de referencia están a la derecha, la región semilla debe ser un rectángulo que se extiende a la izquierda. Uno puede dibujar ambas regiones candidatas usando el mismo conjunto de

ecuaciones (1) simplemente modificando el valor de proporción de los lados más cortos, *h* para que los lados se extiendan más hacia la derecha (más grande *h*) o hacia la izquierda (*h* negativo). En este trabajo se optó por ajustar $h = \pm 4$ (y un pequeño desplazamiento hacia la derecha con *h* negativo), tomando en cuenta las dimensiones de todos los billetes mexicanos (para otros billetes del mundo deben considerarse las proporciones correspondientes). Las regiones producidas se ilustran en la Fig. 4. Claramente, una de las regiones rectangulares (línea roja punteada), que se extiende hacia el lado izquierdo de la denominación contiene sólo los píxeles del billete de banco mientras que la otra región rectangular (línea continua azul) contiene píxeles del billete y algunos del fondo (en este ejemplo se trata de pasto).

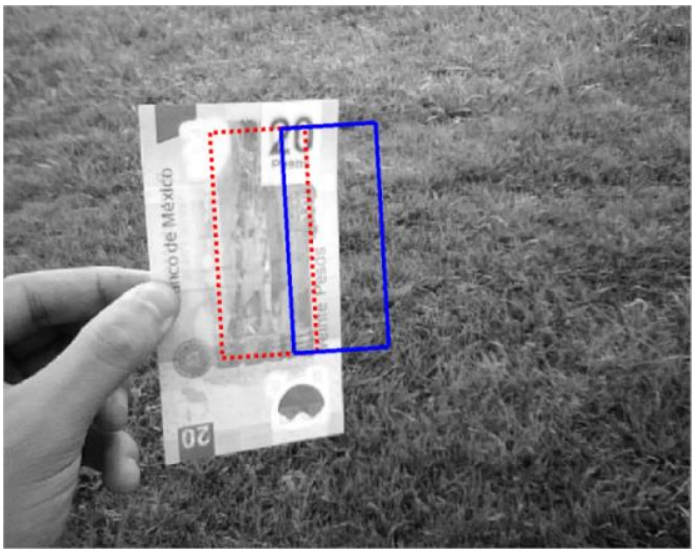


Fig. 4. Regiones rectangulares potencial para convertirse en la Región de Semillas

Con el fin de determinar cuál de las dos regiones candidatas contiene sólo los píxeles del billete, se puede calcular el histograma de color de cada región candidata a comparar con el histograma de la muestra inicial de color I (para hacer factible el cómputo se utiliza un espacio de color RGB reducido a $16^3 = 4096$ colores. Utilizamos el espacio RGB por simplicidad de procesamiento y porque es usado en trabajos similares [7]). Esta comparación (paso 4 en la Fig. 2) se puede conseguir a través de varias funciones de similitud o distancia; en este trabajo se emplea la medida χ^2. Si H_I, H_R y H_L representan los histogramas de la muestra inicial de color *I*, la región candidata que se extiende a la derecha *R*, y la otra región candidata que se extiende hacia la izquierda *L*, la región semilla se determina por:

$$S = argmax_{d\in\{R,L\}} = \{Similitud(H_I, H_d) = e^{-\gamma\chi^2(H_I,H_d)}\} \quad \text{con} \quad \chi^2(H_I, H_d) = \sum_p((H_I - H_d)^2/(H_I + H_d)). \tag{2}$$

El valor del parámetro γ controla la forma de la función exponencial, pero no afecta en la decisión de la región semilla (en este trabajo asignamos γ = 3). La región semilla S obtenida con estos cálculos se utilizará en etapas posteriores.

3.2. Generación de una imagen de similitud de color

La siguiente etapa de nuestra metodología propuesta (paso 5.A en la Fig. 2) consiste en producir una imagen de similitud de color basado en la región semilla obtenida de acuerdo a lo descrito en la sección anterior. Con este fin, hemos optado por utilizar un modelo estadístico paramétrico, dado por una distribución Gaussiana en el espacio de color RGB. Geométricamente, este modelo se puede visualizar como un elipsoide en el espacio RGB, que es una noción útil que se usa en esta exposición. La elección de este modelo se apoya en otros trabajos en la literatura que indican que, para una gran variedad de aplicaciones, ofrece una precisión similar a otros modelos pero tiene menor complejidad [10, 11].

Los parámetros de nuestro modelo estadístico se calculan usando la región semilla, que es una muestra de n píxeles de la región del billete $S = \{S_1, S_2, \dots, S_i, \dots, S_n\}$. El centro del elipsoide corresponde al valor medio de la muestra de color en el espacio RGB dado por:

$$\boldsymbol{\mu}_S = \frac{\sum_{i=1}^{n} S_i}{n}. \tag{3}$$

La orientación y el tamaño del elipsoide está dada por la matriz de covarianza $\boldsymbol{\Sigma}_S$ en sus componentes rojo, verde y azul de cada color en S:

$$\boldsymbol{\Sigma}_S = \begin{pmatrix} \sigma_{RR} & \sigma_{RG} & \sigma_{RB} \\ \sigma_{GR} & \sigma_{GG} & \sigma_{GB} \\ \sigma_{BR} & \sigma_{BG} & \sigma_{BB} \end{pmatrix}, \tag{4}$$

donde σ_{ij}representa la covarianza de los componentes ij.

Una vez que se ha determinado nuestro modelo de color, el siguiente paso en la producción de la imagen de similitud es medir la distancia entre cada píxel de la imagen original y el modelo de color. Dado que las varianzas de los canales rojo, verde y azul en el modelo no son iguales, no sería correcto emplear la distancia Euclidiana tradicional para medir la similitud deseada; la distancia de Mahalanobis [12], $M(\boldsymbol{p}, G(\boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S))$ entre un píxel $\boldsymbol{p}$ y nuestro modelo Gaussiano $G(\boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S)$, es más adecuada para nuestros propósitos, y está dada por:

$$M\big(\boldsymbol{p}, G(\boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S)\big) = \sqrt{(\boldsymbol{p} - \boldsymbol{\mu}_S)^T \boldsymbol{\Sigma}_S^{-1} (\boldsymbol{p} - \boldsymbol{\mu}_S)}. \tag{5}$$

La distancia de Mahalanobis puede convertirse en una medida de similitud si se normaliza entre $[0, 1]$ y se resta de 1. Por lo tanto, nuestra medida de similitud está dada por: $1 - M_N(\boldsymbol{p}, G(\boldsymbol{\mu}\_S, \boldsymbol{\Sigma}\_S))$, donde M_N representa la distancia normalizada de Mahalanobis. El cálculo de esta similitud para cada uno de los píxeles de la imagen original (paso 5.A en la Fig. 2) produce una imagen de similitud de color, denotado por Y. Esta imagen de similitud de color, modificada por una función de peso, se utiliza para obtener la Región De Interés (RDI) final. La finalidad de la función de peso es restringir la forma de la región segmentada correspondiente al billete en caso de que los píxeles en el fondo de la escena presenten colores muy similares a los del billete.

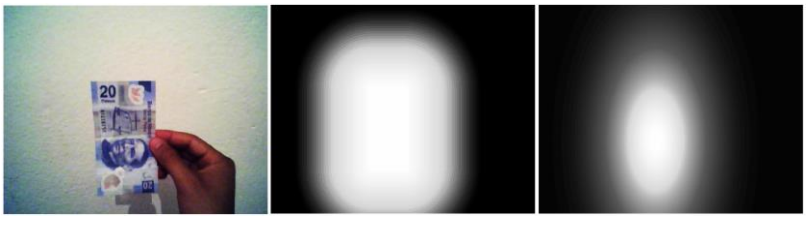
En este trabajo se consideraron dos formas para la función de peso, una rectangular y una elipsoidal. Esto se describe en la siguiente sección.

3.3. Segmentación basada en similitud de color

El modelo de color puede representar efectivamente a los colores de un billete de banco. Sin embargo, todavía existe la posibilidad de que colores muy similares aparezcan en el fondo de la escena, lo que produciría una segmentación incorrecta. Con el fin de reducir esta posibilidad, se emplea una función de peso que actúa como una restricción geométrica (paso 5.B en la Fig. 2). En este trabajo se explora el uso de una función rectangular y una función elipsoidal en el espacio de la imagen como función de peso. Los parámetros de forma de la función de peso, ya sea rectangular o elipsoidal, se pueden determinar de forma automática con base en el tamaño, la posición y orientación de la región semilla, *S*, extraída en la Sección 3.1.

En el caso de la función rectangular, se calcula la transformada de distancia Euclidiana de la región binaria *S* (donde los píxeles en el interior de la región toman el valor de 1 y aquellos en el exterior toman el valor 0). Para cada píxel en *S*, esta transformada asigna la distancia entre ese píxel y el píxel más cercano con valor distinto de cero. En el caso de la función elipsoidal, es necesario como paso intermedio calcular los parámetros de forma (ejes de la elipse, orientación y el centro); esto se logra por medio de un ajuste de mínimos cuadrados a la región semilla *S*. Los parámetros de forma corresponden a los parámetros de una función Gaussiana bidimensional, que al ser normalizada en el rango [0, 1] nos dan una medida de similitud que tal como se hace en el caso de la distancia de Mahalanobis, puede convertirse en una distancia. Las funciones de peso consideradas se ilustran en la Fig. 5.

Cuando la función de peso se aplica a la imagen de similitud de color (restando la imagen de peso de la imagen de similitud), el efecto es que los píxeles que muestran una alta similitud de color pero que se encuentran lejos de la región del billete, tendrán su valor de similitud reducida, lo que limita de manera efectiva su incorporación a la región de interés segmentada. Este criterio de segmentación ha sido utilizado anteriormente con buenos resultados en [7], sin embargo, nuestra propuesta de implementación es mucho menos compleja. El objetivo detrás de la generación de la imagen de similitud de color modificada por una función de peso geométrica es que permita aplicar un valor de umbral para producir la segmentación deseada (paso 6 en la Fig. 2). Se sigue esta estrategia para simplificar la segmentación, de modo que pueda ser usada en el sistema para ayuda a invidentes que requiere funcionar en tiempo real.



a) **b)** **c)**

Fig. 5. a) Imagen de ejemplo; b) función de peso rectangular; c) función de peso elipsoidal. Se ha exagerado un poco el contraste de las imágenes para propósitos de visualización

4. Resultados experimentales

El método de segmentación de billetes descrito anteriormente fue probado en un conjunto de 926 imágenes repartidas entre distintas condiciones que incluyen 3 tipos diferentes de iluminación (a saber, incandescente, fluorescente y luz natural), diferentes denominaciones ($20, $50, $100, $200 y $500), de las caras frontal y posterior del billete y diferente condición (buena condición y deteriorado). Todas las imágenes fueron adquiridas con una cámara Microsoft LifeCam VX-800 con resolución VGA a una distancia aproximada de 40 cm (una posición natural para visualizar un objeto). Con el fin de probar solamente las etapas correspondientes a la segmentación basada en el color, los puntos de referencia de la denominación de billetes se obtuvieron y se validaron previamente [2]. De esta manera, garantizamos que los resultados de las pruebas son independientes de la precisión de la etapa previa. Para la evaluación, la segmentación final generada automáticamente se compara contra anotaciones manuales del conjunto de 926 imágenes, que son referidas como *Ground-Truth*. En la fase experimental se realizaron 3 validaciones: validación de la región semilla generada, del modelo de color y de la segmentación final.

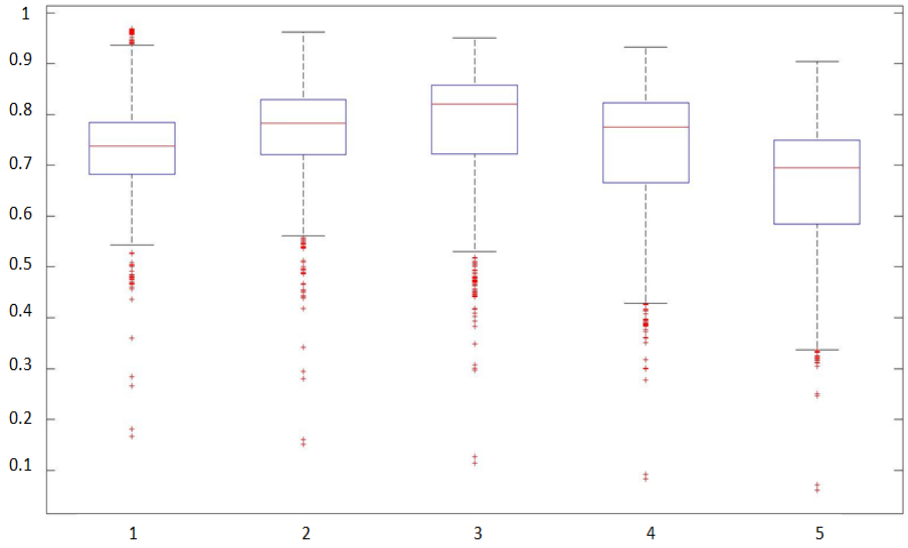


Fig. 6. Resultados de segmentación final con distintos valores de umbral (0.4, 0.45, 0.55, 0.60, 0.65 de izquierda a derecha) con una función de peso Rectangular. El diagrama de cajas nos muestra que se logra una segmentación promedio de 82% para el valor de umbral de τ = 0.55 como mejor resultado de segmentación.

Validación de la región semilla - Basados en los puntos de interés obtenidos por medio de la transformada FRS, regiones rectangulares centradas fueron producidas y una región semilla fue seleccionada de acuerdo con el procedimiento descrito en la Sección 3.1. El objetivo de este experimento es cuantificar la proporción de imágenes para las que el método propuesto tiene éxito. El criterio de validación se basó en el área de intersección, en píxeles, entre el *GroundTruth* y la región semilla (*SeedRegion*),

dividida por el área total de la región semilla: $N(GroundTruth \cap SeedRegion)/N(SeedRegion)$, donde $N(\cdot)$ representa el número de píxeles dentro de una región dada. Se consideró que la Región semilla se produce correctamente cuando al menos una proporción del 85% de dicha región se encuentra en el $GroundTruth$ del billete. Como resultado de esta prueba, se encontró que aproximadamente el 98% de las imágenes satisfacen esta condición de validez.

Validación del modelo de color - Después de haber obtenido la región semilla para cada imagen de los billetes, se obtuvieron los parámetros necesarios para definir nuestro modelo de color como se describe en la Sección 3.2. Con el fin de cuantificar la validez de los parámetros calculados, los valores producidos de forma automática se compararon con los obtenidos con base en los píxeles que corresponden al *Ground Truth*. La raíz del error cuadrático medio para las medias, por canal, es: Rojo: 6.81; Verde: 7.88; Azul: 6.45. La raíz del error cuadrático medio para las desviaciones estándar, por canal, es: Rojo: 4.53; Verde: 3.35; Azul: 5.33.

Validación de la segmentación final - Utilizando los valores de los parámetros del modelo de color que se estimaron a partir de la región semilla, se calculó la distancia de Mahalanobis entre cada píxel de la imagen original y del modelo y se produjo una imagen de similitud a través del procedimiento descrito en la Sección 3.2. Se establecieron distintos valores de umbral de decisión τ, y se obtuvo una RDI final por cada imagen. Con el fin de cuantificar la precisión de la RDI segmentada, se obtuvo el coeficiente de Tanimoto [13] entre la RDI y el *Ground Truth* correspondiente. Además dichas regiones se compararon entre sí, de manera similar como se hizo en la validación de la región semilla (Sección 4.2). Los resultados para una función de peso rectangular se muestran en la Fig. 6 y para la función de peso elipsoidal en la Fig. 7. En la Tabla 1 se muestra los datos resultantes de la evaluación realizada. Un ejemplo del resultado contra la anotación manual se ilustra en la Figura 8.

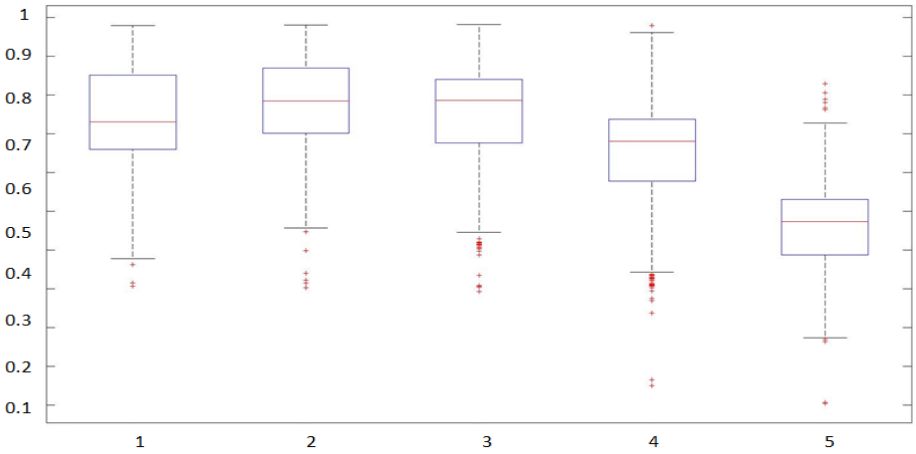


Fig. 7. Resultados de segmentación final con distintos valores de umbral (0.75, 0.80, 0.85, 0.90 de izquierda a derecha) con una función de peso Elipsoidal. El diagrama de cajas nos muestra que se logra una segmentación promedio de 78% para el umbral de τ = 0.80 como mejor resultado de segmentación

Tabla 1. Resultado de la comparación entre la segmentación automatica y el Ground Truth

Medida	**Promedio ± Std. Dev.**
Coeficiente de Tanimoto	0.82 ± 0.11
Parámetro	**Error Cuadrático Medio**
Promedio RGB	R: 4.64 G: 4.82 B: 5.57
Std. Dev.	R: 3.53 G: 2.89 B: 4.92

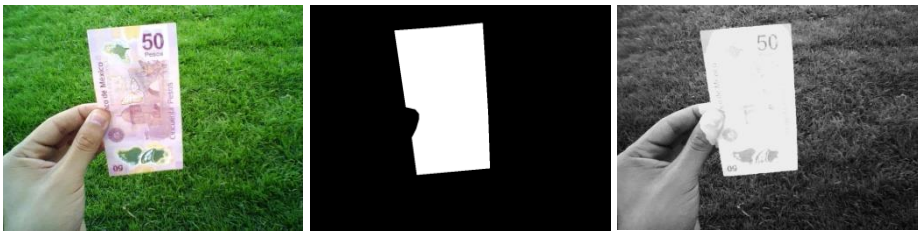


Fig. 8. De izquierda a derecha: Ejemplo de un billete, su anotación manual y la Región de Interés final (región resaltada) obtenida de forma automática

5. Discusión

Las pruebas que se llevaron a cabo indican que una representación basada en conjuntos de histogramas de color es eficiente para distinguir entre regiones y así lograr la segmentación de imágenes. Sólo el 2% de las regiones de referencia extraídas de forma automática (Regiones Semilla) no cumplieron con los criterios de validez establecidos. Por otra parte, la comparación entre los parámetros de los modelos de color obtenidos a partir de las Regiones de semillas y las correspondientes a las regiones *Ground Truth* muestran que estos conjuntos de parámetros son bastante cercanos uno del otro; en la etapa final de segmentación se muestra que el modelo de color es útil para obtener automáticamente RDIs (los billetes de banco), ya que el coeficiente de Tanimoto promedio es de 82%. Por otra parte, los resultados presentados con las dos funciones de peso muestran que usar la función rectangular produce un mejor porcentaje de segmentación, con 82%, siendo superior al 78% producido mediante el uso de la función elipsoidal.

Cabe recordar que el objetivo final del método propuesto en este trabajo es definir una región de la que se pueda obtener una buena muestra de color, ya que esta información será utilizada posteriormente para la clasificación robusta por color de los billetes. La confirmación de que el procedimiento presentado en este trabajo alcanza dicho objetivo, proviene de los resultados de la comparación que se lleva a cabo entre el conjunto de los parámetros del modelo obtenidos de las RDIs producidos de forma automática y el conjunto de parámetros de los modelos obtenidos a partir de las regiones *Ground Truth* (posiblemente la mejor estimación de los parámetros reales); respectivamente, se observó que la raíz del error cuadrático medio entre la media estimada y la verdadera media es de 4.64, 4.82 y 5.57 para los canales rojo, verde y

azul. Esto significa que la distancia media entre la media estimada y la media real en el espacio RGB es 6.33 ± 5.98, o que aproximadamente el 85% de los valores estimados se encuentran a una distancia de menos de 10 a partir de la mejor estimación en el espacio RGB de 8-bits, que consideramos un muy buen resultado, dado el tamaño del espacio RGB. Adicionalmente, resultados preliminares de clasificación han sido positivos, apoyando nuestra evaluación del método; esto se discutirá en un trabajo futuro.

6. Conclusiones y trabajo futuro

Se presentó un método para la segmentación de los billetes mexicanos en imágenes digitales basado en el color. En este método, una vez que los puntos de interés de la denominación del billete de banco se han extraído, se selecciona una región de la semilla y de esto, se obtiene un modelo de color. La segmentación final se obtiene mediante el cálculo de una imagen de similitud, donde cada píxel de la imagen original se le asigna un valor de distancia al modelo computarizado. El método de segmentación propuesto ha sido desarrollado como parte de un sistema de asistencia para discapacitados visuales. Actualmente estamos trabajando en una idea similar para lograr la clasificación automatizada de billetes de banco, que completará el módulo de reconocimiento de billetes de nuestro sistema de ayuda. En el futuro, nuestro objetivo es extender el método propuesto a billetes de otros países; esto es posible ya que muchos países en el mundo adoptan esquemas para que su moneda pueda distinguirse principalmente por su color y por numerales impresos en una fuente de buen tamaño con el fin de ayudar a personas con debilidad visual, y a máquinas, a distinguirlos entre sí.

Agradecimientos. El presente trabajo se desarrolló con apoyo del Consejo Nacional de Ciencia y Tecnología de México, a través de los programas: CATEDRAS-2598 (A. Rojas) y el Programa de Becas Nacionales (331907- J. Flores, 331797- R. López).

Referencias

1. AMFECCO, sitio web: http://www.amfecco.org/article_estadisticas.php.
2. Rojas-Domínguez, A., Lara-Alvarez, C., Bayro-Corrochano, E.: Automated banknote recognition for the visually impaired. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science, 8827, pp. 572–579 (2014)
3. García-Lamont, F., Cervantes, J., López, A.: Recognition of Mexican banknotes via their color and texture features. Expert Systems with Applications, Vol. 39, No. 10, pp. 9651–9660, ISSN 0957-4174 (2012)
4. Mohamed, A., Ishak, M.I., Buniyamin, N.: Development of a Malaysian Currency Note Recognizer for the Vision Impaired. In: Engineering and Technology (S-CET), Spring Congress on, pp.1–4, 27-30 May (2012)
5. Solymár, Z., Stubendek, A., Radványi, M., Karacs, K.: Banknote Recognition for Visually Impaired. In: European Conference on Circuit Theory and Design (ECCTD) 20th, Budapest, Hungary, pp. 841–844 (2011)

6. Wang, S: Color Image Segmentation Based on Color Similarity. In: Computational Intelligence and Software Engineering. CiSE 2009, International Conference on, pp.1–4, 11-13 Dec. (2009)
7. Ning, J., Zhang, L., Zhang, D., Wu, C.: Interactive image segmentation by maximal similarity based region merging. Pattern Recognition, Vol. 43, No. 2, February, pp. 445–456, ISSN 0031-3203 (2010)
8. Williams, M.M., Anderson, R.G.: Currency design in the United States and abroad: Counterfeit deterrence and visual accessibility. Federal Reserve Bank of St. Louis Review, Sept-Oct, pp. 371–414 (2007)
9. Loy, G., Zelinsky, A.: Fast radial symmetry for detecting points of interest. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 8, pp. 959–973 (2003)
10. Lee, J.Y., Yoo, S.I.: An elliptical boundary model for skin color detection. In: Proceedings of the International Conference on Imaging Science, Systems and Technology (2002)
11. Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. Pattern Recognition, Vol. 40, No. 3, pp. 1106–1122 (2007)
12. Prasanta Chandra, M.: On the generalised distance in statistics. In: Proceedings of the National Institute of Sciences of India, Vol. 2, No. 1, pp. 49–55. Retrieved 2012-05-03 (1936)
13. Crum, W., Camara, O., Hill, D.: Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis. IEEE transactions on medical imaging, Vol. 25, No. 11, pp. 1451–1461 (2006)

Detección de obstáculos durante vuelo autónomo de drones utilizando SLAM monocular

José Martínez-Carranza[1], Luis Valentín[1], Francisco Márquez-Aquino[1], Juan Carlos González-Islas[1], Nils Loewen[2]

[1] Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, México

[2] Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Queretaro, Queretaro, México

carranza@inaoep.mx,luismvc@ccc.inaoep.mx, A00889524@itesm.mx

Resumen. En este artículo se describe una metodología para la detección de obstáculos que aparecen de manera repentina durante el vuelo autónomo de un dron. Para lograr lo anterior, este trabajo se basa en el uso de un sistema de localización y mapeo simultaneo o bien, Simultaneous Localisation and Mapping (SLAM) en inglés, monocular. El sistema de SLAM monocular es utilizado para obtener estimaciones de la posición del dron así como un mapa conformado por puntos 3D que representan el ambiente que se observa a través de la cámara a bordo del vehículo. Es importante resaltar que ningún otro sensor de profundidad, de medición de distancia o inercial es utilizado en este trabajo, y que dicho sistema puede ser utilizado de manera indistinta cuando el dron vuela en espacios interiores o exteriores, puesto que la localización del dron y el mapa del ambiente no dependen de la señal GPS o de algún sistema de localización externo, sino únicamente del procesamiento de las imágenes capturadas por la cámara.

Palabras clave: Vuelo autónomo, detección de obstáculos, drones, SLAM monocular.

Obstacle Detection during Autonomous Flight of Drones Using Monocular SLAM

Abstract. We describe a methodology for the detection of obstacles suddenly appearing during the autonomous flight of a drone. For this, our approach is based on the use of a monocular Simultaneous Localisation and Mapping (SLAM) system. The monocular SLAM system is used to obtain drone's position estimates, as much as to generate a 3D map of the scene, which is observed with an on-board camera. It is important

to highlight that no other sensor is used in this work, and that our approach works in both, indoor and outdoor scenes, due to the fact that the drone's localisation does not depend upon GPS signal or any other external localisation system, but only on the processing of the imagery captured by the on-board camera.

Keywords: Autonomous flight, obstacle detection, drones, monocular SLAM.

1. Introducción

En los últimos años, la robótica ha sido utilizada con gran interés en diferentes ámbitos, incluyendo aquellos en los que se desarrollan aplicaciones civiles. Algunas organizaciones como DARPA (del inglés, Defense Advanced Research Projects Agency) y RoboCup han impulsado el desarrollo de tecnología para robots móviles autónomos, como lo son humanoides, robots de servicio y vehículos aéreos no tripulados (VANTs) [16], también conocidos como drones.

Recientemente, el uso de drones ha tenido un crecimiento exponencial en la investigación y en el ámbito comercial con un número creciente de aplicaciones, por ejemplo: búsqueda y rescate, vigilancia, inspección aérea de estructuras, monitoreo agroindustrial y forestal, videografía, entre otras. Dichas actividades se realizan tanto en ambientes exteriores como interiores densamente poblados y con un alto dinamismo en el dominio espacio-temporal. Por lo que se requiere que un dron vuele eficientemente de manera autónoma con las capacidad de detección y evasión de obstáculos [3]. En ese sentido, la información obtenida por los sistemas de visión a bordo, mejora significativamente las capacidades del dron para realizar dichas tareas. El uso de cámaras como único sensor extereoceptivo en los drones y en los robots en general, provee de información fotométrica detallada que se utiliza por diferentes algoritmos, los cuales determinan la inteligencia del robot para realizar tareas de localización, navegación, seguimiento y manipulación con mayor eficiencia y robustez [18].

El vuelo autónomo es una capacidad deseada en los drones por diversas razones. Discutiblemente, quizás la más importante es que el dron pueda tomar control de su propio vuelo en caso de que el piloto pierda comunicación con el vehículo, en cuyo caso sería deseable que el dron tuviera la capacidad de identificar dicho evento y por lo tanto volar de manera autónoma hacia algún punto de aterrizaje fuera de riesgo. Actualmente, la solución para lograr un vuelo autónomo es a través de un GPS, el cual trabaja muy bien en exteriores y en áreas libres de obstáculos. Sin embargo, este tipo de sistemas son muy propensos a fallar en ambientes de interiore, dinámicos y densamente poblados. Por lo que, aplicaciones de vigilancia en este tipo de ambientes no podrían ser realizadas con esta solución [10]. Por otra parte, aún cuando el vuelo autónomo se ejecutara con soporte GPS, ésto no libra al vehículo del riesgo de colisionar con objetos que se atraviesen en medio de la ruta de vuelo, por ejemplo: aves, cables, antenas o incluso personas (cuando el vehículo se acerque a tierra).

Motivados por lo anterior, en este trabajo se propone una metodología para realizar navegación autónoma de un dron con la capacidad de detección de obstáculos. Para esto, se utiliza el sistema SLAM monocular propuesto en [12], para estimar la posición del vehículo; y simultáneamente obtener un mapa de puntos en $\mathbb{R}^3$ que representan su entorno. La detección del obstáculo se hace mediante el análisis de las imágenes capturadas con la cámara a bordo. El algoritmo para corregir la posición y controlar la velocidad del dron durante la navegación y detección de obstáculos está basado en un controlador Proporcional Integral Derivativo (PID). Es importante resaltar que ningún otro sensor (por ejemplo, inercial o de profundidad), es utilizado para detectar el obstáculo. Para la evaluación de la metodología propuesta, se utiliza un sistema óptico de captura de movimiento en 3D (Vicon®), con el cual se obtiene la posición del dron. Dicho sistema se usa para medir con precisión la distancia a la cuál el algoritmo de control frena y aleja al dron cuando el obstáculo es detectado. Los resultados obtenidos indican que la propuesta aquí presentada es factible y eficaz.

Con el objetivo de describir este trabajo, este artículo ha sido organizado como se indica a continuación: la Sección 2 presenta el trabajo relacionado; la Sección 3 describe la metodología propuesta; los experimentos y resultados se discuten en la Sección 4; las conclusiones y trabajo futuro se presentan en la Sección 5.

2. Trabajo relacionado

Una de las tareas principales cuando se navega de manera autónoma con un dron es la detección y evasión de obstáculos. Este tema ha sido investigado ampliamente en aplicaciones para sistemas automáticos de control de tráfico aéreo, en donde se requiere que el vehículo identifique obstáculos en su trayectoria y haga una replaneación de la ruta [1]. En ese sentido, la detección de obstáculos, mediante el uso de un telémetro láser, y la replaneación dinámica en helicópteros no tripulados, se han estudiado por Scherer et al. [14] and Shim et al. [15].

El uso de flujo óptico y sensores inerciales ha permitido realizar rutinas de navegación, como son: estimación de distancia [7], evasión de obstáculos [19], [11], estimación de velocidad y altura [5], entre otros. Sin embargo, el flujo óptico acumula error en estimación de trayectorias largas.

El uso de algortímos de visión artificial en los drones ha incrementado de manera sustancial las capacidades del sistema para realizar tareas de navegación. Watanabe [17] aborda el problema de evasión de obstáculos con un helicóptero no tripulado a través de un sistema de visión monocular. Aunque esta propuesta no entrega mediciones absolutas de distancia de los obstáculos, presenta buenos resultados en un ambiente simulado.

Por otra parte, si el movimiento de la cámara es conocido, se puede usar SFM (Del inglés, Structure From Motion) a través del seguimiento de puntos de interés para medir la profundidad de la escena, tal como lo proponen Call et al. [2]. En ese mismo sentido, Hrabar [8] propone el uso de visión estéreo para la

evasión de obstáculos. Dicha técnica ha sido usada previamente para estimación de movimiento y altura en [13].

3. Descripción del sistema

En este trabajo se utiliza un vehículo aéreo comercial de bajo costo de la empresa Parrot, el cual dispone de un SDK (Del inglés, *Software Development Kit*) para desarrollo. El dron tiene una duración de carga para vuelo de 10 minutos. Además, este dron cuenta con una cámara abordo con la cual se puede capturar video con una resolución de 1920 x 1080 píxeles a 30 cuadros por segundo; el video se transmite de manera inalámbrica (a través de WiFi) a la estación terrestre para ser procesada. La Figura 1 muestra una vista en perspectiva del vehículo.

Fig. 1. Vehículo aéreo Bebop. Figura tomada de http://blog.parrot.com

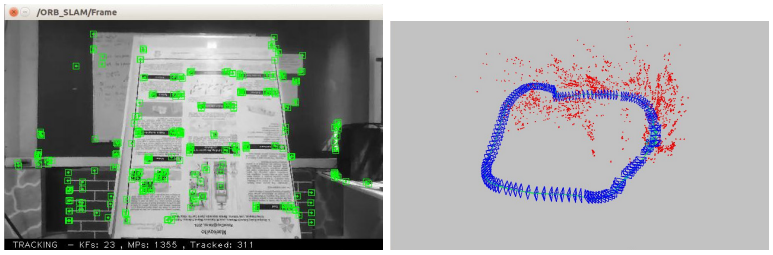
El procesamiento de imágenes y datos para la navegación autónoma y la evasión de obstáculos del dron no se hace a bordo. Para ello se utiliza una computadora portátil con un procesador a 64 bits Intel Core i7-4720HQ a 2.60 GHz x 8, y una memoria RAM de 16Gb.

El sistema se compone de tres etapas, las cuales son integradas con ROS (Del ingles, Robotic Operative System). La primera etapa consiste en la adquisición de una secuencia de imágenes a través de la cámara del vehículo. La secuencia de imágenes adquiridas es la entrada del algoritmo ORB-SLAM [12] empleado en este trabajo (ver sub-seccin 3.1), el cual permite al sistema móvil localizarse y generar un mapa de puntos en $\mathbb{R}^3$ de su entorno, este proceso conforma la segunda etapa del sistema. La tercera etapa es el sistema de control, el cual controla al dron para navegar de manera autónoma y detectar obstáculos. La medición de la distancia a la que el vehículo se detiene para detectar el obstáculo es realizada con el sistema Vicon®.

3.1. ORB-SLAM

El algoritmo ORB-SLAM es un sistema monocular en tiempo real de localización y mapeo simultáneo basado en la detección de puntos de interés

representados por descriptores ORB (del inglés, Oriented FAST and Rotated BRIEF). El sistema es robusto a movimientos erráticos, provee las capacidades de relocalización y cierre de trayectoria, así como una inicialización automática de los puntos del ambiente. La Figura 2(a) muestra la imagen de correspondencia del seguimiento de descriptores en el ambiente real observado por la cámara. Mientras que la Figura 2(b) ilustra la representación del mapa de puntos en $\mathbb{R}^3$ del ambiente y la localización de la pose de la cámara estimada a través del algoritmo ORB-SLAM. Nótese que en esa figura también se observa el cierre de trayectoria, característica importante del ORB-SLAM.



(a) Imagen de correspondencia de los descriptores en el entorno real empleando ORB-SLAM.

(b) Mapa y localización empleando ORB-SLAM. Los puntos rojos son los puntos en $\mathbb{R}^3$ del ambiente y los cuadros azules representan las poses de la cámara.

Fig. 2. Sistema ORB SLAM en ejecución

El algoritmo ORB-SLAM tiene tres componentes principales, el seguimiento, el mapeo local y el cierre de trayectoria usando descriptores visuales ORB.

La etapa de seguimiento es la encargada de estimar las poses de la cámara mediante la búsqueda de correspondencias de los puntos de interés del mapa entre los *frames* anteriores y el *frame* actual. Además, la etapa de seguimiento también toma la decisión de cuando insertar un *keyframe* para actualizar el grafo basándose en el cambio de información visual entre el *keyframe* actual y los anteriores. El componente de mapeo local es el que agrega nuevos puntos al mapa y los mantiene. La optimización del mapa construido se hace mediante *Bundle Adjustment*[6]. Finalmente, la etapa del cierre de trayectoria se emplean cuando un lugar es revisitado. Una vez que el sitio es detectado, se utiliza optimización global para actualizar el mapa. En la Figura 3 se ilustra un diagrama a bloques del sistema ORB-SLAM. Si se requiere información detallada de dicho sistema consulte [12].

3.2. Sistema de control

El sistema de control es la etapa encargada de mantener el vuelo autónomo del dron entre dos puntos de referencia. Esto se logra mediante la corrección

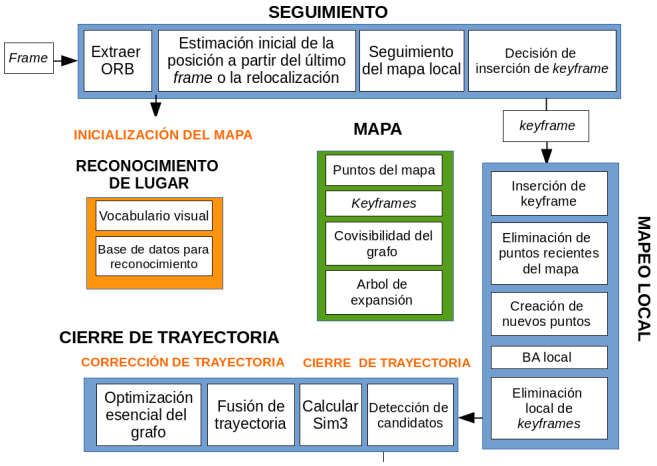


Fig. 3. Panorama general del algoritmo ORB-SLAM. Este diagrama fue adaptado al español de [12]

del ángulo de elevación (*"pitch"*) y de dirección (*"yaw"*) del vehículo a través de un controlador proporcional integral derivativo (PID). El controlador PID es una de las formas más comunes de control por retroalimentación de un sistema en donde se desea reducir el error entre un valor medido y una referencia. La ecuación de un controlador PID se describe mediante la Ecuación 1:

$$u(t) = k_p e(t) + k_i \int_o^t e(\tau)d\tau + k_D \frac{de(t)}{dt}. \quad (1)$$

En la Ecuación 1, u es la señal de control, $e = y_{esp} - y$, es el error de control, y es la variable medida, y_{esp} es la señal de referencia.

La señal de control es la suma de tres términos, el componente proporcional al error, el componente proporcional a la integral del error y el componente proporcional a la derivada del error, en cada uno de estos términos se definene las constantes k_p, k_Dy k_i respectivamente. Las constantes k_p, k_Dy k_i son determinadas experimentalmente En nuestro caso, la referencia del controlador PID es dinámica y se calcula con la distancia mínima entre el promedio de los puntos del mapa y la posición del vehículo, sobre el eje en el que el dron se desplaza. En este caso se emplea como referencia un umbral de 3/4 de la distancia mínima.

En la Figura 4 se muestra un diagrama de flujo en el que se observa el esquema de control y los bloques que conforman el detector de obstáculos. Puede apreciarse que hay dos controles PID, uno que mantiene al vehículo desplazándose hacia el frente, siempre que no haya algún obstáculo en la trayectoria del vehículo, y otro que lo hace retroceder en caso de que la detección de un obstáculo sea positiva; el vehículo retrocederá hasta que éste regrese al origen de coordenadas para así reiniciar nuevamente el vuelo hacia adelante.

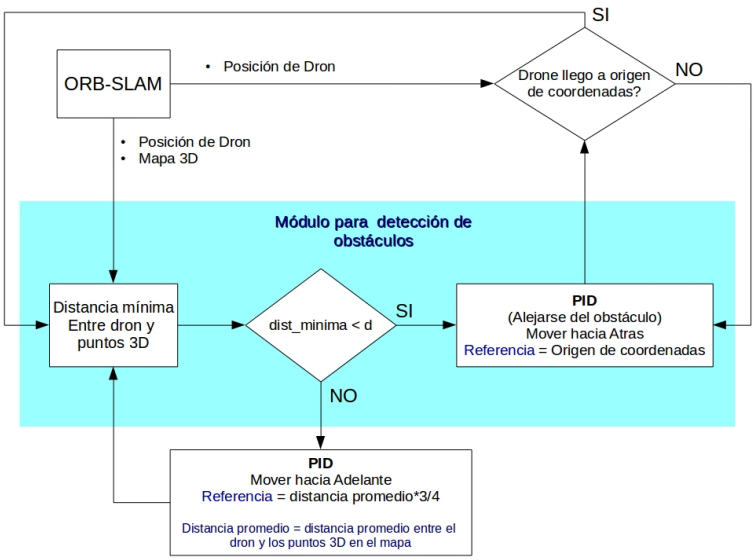


Fig. 4. Diagrama de flujo del esquema de detección de obstáculos propuesto. En el cuadro azul claro se indican los módulos responsables de la detección de un obstáculo cercano al vehículo

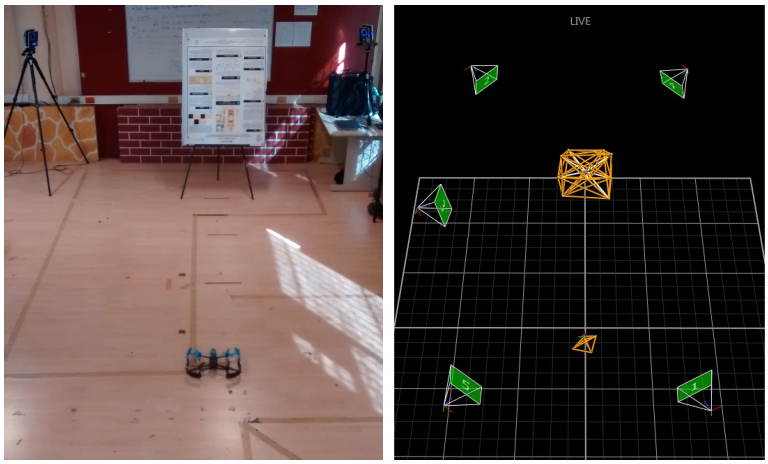
4. Resultados y experimentos

Para evaluar el funcionamiento de la detección de obstáculos se realizaron dos tipos de experimentos. En el primer experimento se realizó una comparación de dos trayectorias arbitrarias entre el sistema Vicon®y el sistema de localización y mapeo ORB-SLAM. En el segundo experimento se evaluó el desempeño de la detección de obstáculos, tomando como *sistema de referencia* el sistema externo de localización Vicon®. Para este experimento, se realizaron 40 pruebas en donde el vehículo se desplazó sobre una trayectoria recta y se varió la posición del obstáculo.

Los experimentos se realizaron en un ambiente interior, con una área de trabajo de aproximadamente 6.5m X 3m (largo X ancho) delimitada virtualmente por el área de cobertura del sistema de seguimiento. Las condiciones de iluminación fueron controladas. La Figura 5 describe la configuración del escenario.

4.1. Experimento 1: Comparación de sistemas de localización

En este experimento se comparó la trayectoria obtenida con ORB-SLAM contra el sistema de localización Vicon®para determinar la precisión del sistema SLAM. Es importante considerar que el sistema ORB-SLAM es egocéntrico, lo



(a) Imagén real del área de trabajo. En la imagen se observa el dron en el origen del mundo.

(b) Representación del área de trabajo obtenida por el sistema de seguimiento. En la imagen se observa la distribución de las cámaras (en verde), el dron y el obstáculo.

Fig. 5. Área de trabajo: en ambas imágenes se aprecia el obstáculo en la parte superior y el dron en la parte inferior

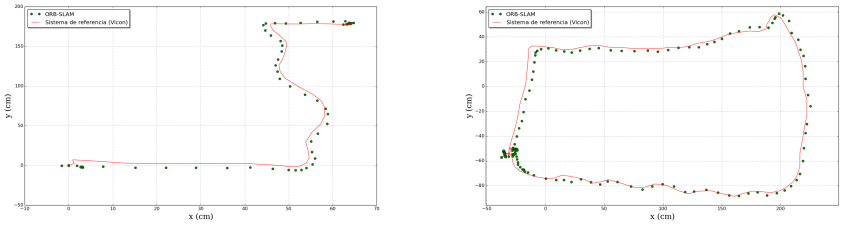
cual significa que el origen de referencia se ubica en el punto donde inicia el sistema, por lo tanto, para compara las trayectorias, se realizó una transformación sobre el marco de referencia del sistema ORB-SLAM de tal forma que coinsidiera con el marco de referencia del sistema de seguimiento. Por otro lado, la trayectoria obtenida con el sistema ORB-SLAM tiene una escala arbitraria adimensional, por lo que fue necesario realizar un escalamiento para compararla con el sistema de referencia. ORB-SLAM se ejecutó a una taza aproximada de 30Hz.

Las pruebas para este primer experimento se realizaron de la siguiente forma:

1. Se despegó el vehículo en cada prueba en el mismo punto de partida.
2. Se inició la captura de posiciones del dron con el sistema de referencia Vicon®.
3. Se inició la captura de posiciones del dron con ORB-SLAM.
4. Se condujo al vehículo a través de una trayectoria aleatoria.
5. Se comparan ambas trayectorias para determinar la precisión de la localización de ORB-SLAM.

En este experimento se realizaron dos pruebas en diferentes configuraciones las cuales están representadas en la Figura 6, para cada prueba se calculó el error cuadrático medio o RMS (Del inglés, *Root Mean Square*) entre el ORB-SLAM y el sistema Vicon®. Los resultados detallados se reportan en la Tabla

1. La trayectoria mostrada en la Figura 6(a) es útil para indicar la habilidad de ORB-SLAM de estimar una trayectoria sin restricción en los grados de libertad (traslación y orientación). Por otra parte, la trayectoria mostrada en la Figura 6(b) es útil para determinar el error en términos porcentuales, puesto que el error es de 5 cm aproximadamente (ver Tabla 1) sobre una trayectoria de 800 cm, lo cual implica un error porcentual de menos del 1 %. De este modo, este es el error esperado de un sistema SLAM monocular [4,9] y puede por tanto, ser utilizado con cierto nivel de confianza para estimar la posición del vehículo.



(a) Prueba 1. Trayectoria en forma de L.

(b) Prueba 2. Recorrido en forma de rectángulo con cierre de trayectoria.

Fig. 6. Gráficas del Experimento 1. Seguimiento de una trayectoria aleatoria, donde la trayectoria roja representa el seguimiento del Vicon®y la trayectoria verde representa el seguimiento con el ORB-SLAM

Tabla 1. Resultados del Experimento 1: comparación de sistemas

Resultados del experimento 1	
# Prueba	RMS (cm)
Prueba 1	5.2283
Prueba 2	5.2794

4.2. Experimento 2: Detección de obstáculos

En este experimento se determinó la capacidad del sistema para la detección de obstáculos. Para evaluar esta detección, el vehículo voló de forma autónoma oscilando entre dos puntos de una linea recta. Durante su recorrido un obstáculo se colocó de frente al vehículo obstruyendo el paso. En este experimento dos elementos fueron rastreados por el sistema Vicon®el vehículo y el obstáculo, esto con el fin de calcular la distancia entre ambos elementos. No obstante, es

importante recalcar que la posición del dron obtenida con ORB-SLAM fue la única información utilizada por el control Proporcional Integral Derivativo que se utilizó para el vuelo autónomo.

Este experimento se realizó con tres configuraciones, donde el obstáculo se colocó a diferentes distancias separadas por 1 metro, como se ve en la Figura 7. Cada configuración está representadas por las letras a,b,c, donde a es la configuración más lejana al origen O, con una distancia de 2.75m. Se utilizó un rectángulo sólido de 1.5m x 1m altamente texturizado como obstáculo. En cada configuración se realizaron 10 pruebas.

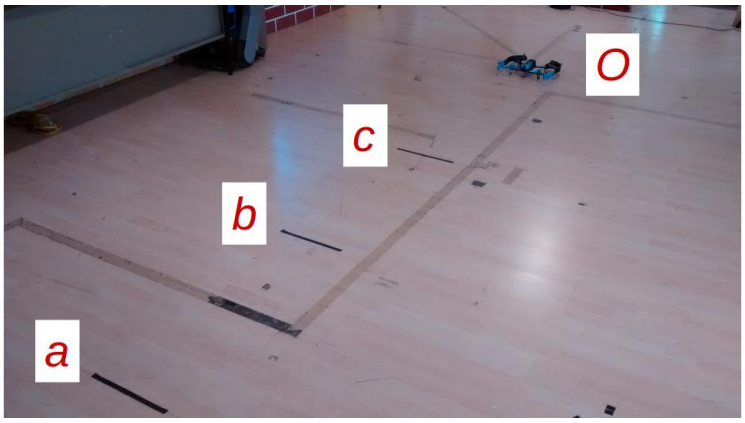


Fig. 7. Configuraciones del Experimento 2. Cada configuración en donde se colocará el obstáculo está representada por las letras a,b,c y O representa el origen del sistema Vicon®

Tabla 2. Tabla de resultados del Experimento 2 para las configuraciones a y b, donde d-obs representa la distancia a la que se coloca el obstáculo con respeto al origen del sistema, d-E representa la distancia promedio de detección es decir la distancia mínima entre el vehículo y el obstáculo

Resultados del experimento 2				
Configuración	d-obs (cm)	d-E (cm)	% detección	% colisión
a	275	39.60	100	0
b	175	45.97	80	20

Las pruebas se realizaron de la siguiente forma para cada configuración:

1. Se despegó el vehículo en cada prueba en el mismo punto de partida (el origen del sistema Vicon®).

Tabla 3. Tabla de resultados del experimento 2 para las configuraciones *c.1* y *c.2*, donde d-obs representa la distancia a la que se coloca el obstáculo con respecto al origen d-E representa la distancia promedio de detección es decir la distancia mínima entre el vehículo y el obstáculo

Resultados del experimento 2				
Configuración	d-obs (cm)	d-E (cm)	% detección	% colisión
c.1	75	31.55	10	90
c.2	75	64.12	100	0

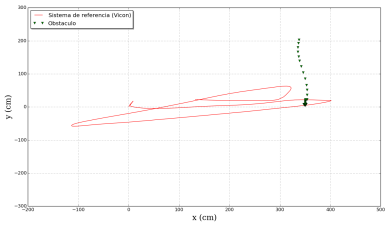
2. De forma autónoma el vehículo se movió hasta la pared frontal y retrocedió hasta la pared trasera.
3. Se inició la captura de posiciones del dron con el sistema de referencia Vicon®.
4. Se inició la captura de posiciones del dron con ORB-SLAM.
5. Se inició el vuelo autónomo.
6. Se obstruyó el paso del vehículo cuando viajaba nuevamente hacia la pared frontal.
7. Se calculó distancia mínima a la que se detiene el vehículo antes de colisionar con el obstáculo.

Se realizaron 40 pruebas con las configuraciones antes descritas. Los resultados de cada configuración fueron resumidos en dos tablas. La Tabla 2 resume los resultados obtenidos en las 20 pruebas de las configuraciones *a* y *b*, donde se utiliza las mismas constantes de control. La Tabla 3 resume 20 pruebas para la configuración *c* donde, *c.1* utiliza las mismas constantes de las configuraciones anteriores y *c.2* emplea constantes reducidas con las que se obtienen mejores resultados. Lo anterior demuestra que el algoritmo de control y detección de obstáculos está sujeto a la sintonización experimental de las constantes del control PID.

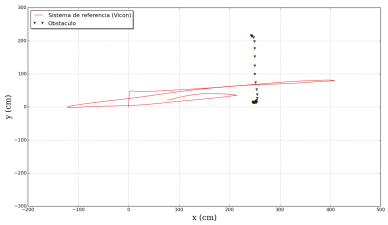
La Figura 8(a) ilustra las trayectorias representativas de cada configuración, donde la linea continua representa la trayectoria del dron y la linea punteada representa la trayectoria del obstáculo. Es importante mencionar que en un principio el obstáculo no se interpone en la trayectoria del vehículo, por tanto se aprecia que el dron tiene una trayectoria más prolongada en el principio, sin embargo, cuando el obstáluco es introducido el sistema es capaz de detectarlo y detener al dron para evitar la colisión.

5. Conclusiones y trabajo futuro

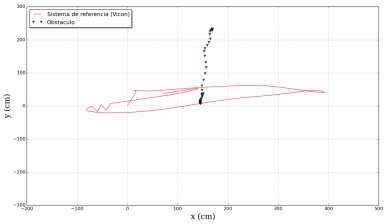
En este trabajo se ha presentado una metodología para la detección de obstáculos durante vuelo autónomo de un dron. Esta metodología se basa únicamente en el procesamiento de las imágenes obtenidas con una cámara a bordo del dron. Estas imágenes son procesadas por un sistema SLAM monocular que



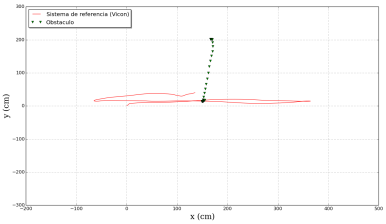
(a) Gráfica representativa de la configuración *a.*



(b) Gráfica representativa de la configuración *b.*



(c) Gráfica representativa de la configuración *c.1.*



(d) Gráfica representativa de la configuración *c.2.*

Fig. 8. Gráficas del Experimento 2, donde se dibujan trayectorias representativas de cada configuración

genera una estimación de la posición de dron así como un mapa de puntos en $\mathbb{R}^3$ que representan al ambiente, observado con la cámara del dron. De este modo, dicha posición es utilizada por un controlador PID para ejecutar el vuelo autónomo del dron y hacer que éste siga una ruta. Así mismo, el mapa también es utilizado para determinar cuando un objeto, descrito por un conjunto de puntos en $\mathbb{R}^3$ en el mapa, aparece en medio de la ruta y por lo tanto, el vuelo debe ser detenido para evitar un impacto contra el objeto.

Adicionalmente, se han presentado resultados que ilustran la efectividad de la metodología propuesta. Con la ayuda de un sistema de localización externo (Vicon), se logró medir la distancia promedio a la que el vehículo logra detenerse a raíz de detectar el obstáculo con un éxito de más del 80 % a una distancia de 40-60 cm aproximadamente del obstáculo, lo cual permite un frenado suave así como el que el control pueda alejar paulatinamente al dron del obstáculo detectado. Lo anterior indica que el procesamiento de las imágenes se realiza a una tasa suficiente, al menos para el escenario experimental planteado.

Finalmente, este trabajo contribuye a demostrar el tipo de capacidades que pueden ser implementadas para que un dron ejecute un comportamiento inteligente, esto es, el de detectar y así evitar chocar con un obstáculo, utilizando una cámara a bordo como único sensor para observar el ambiente. Lo anterior, es atractivo ya que permite pensar en drones de bajo costo y tamaño reducido,

equipados con un número reducido de sensores, lo que su vez ahorra energía y peso.

El trabajo a futuro incluye el escalar la metodología presentada en este trabajo a un escenario en exteriores, para rutinas de vuelo más complejas y con obstáculos de diferentes tamaños y texturas.

Agradecimientos. Este trabajo fue financiado por la Royal Society-Newton Advanced Fellowship con referencia NA140454.

Referencias

1. Albaker, B., Rahim, N.: A survey of collision avoidance approaches for unmanned aerial vehicles. In: technical postgraduates (TECHPOS), 2009 international conference for. pp. 1–7. IEEE (2009)
2. Call, B., Beard, R., Taylor, C., Barber, B.: Obstacle avoidance for unmanned air vehicles using image feature tracking. In: AIAA Guidance, Navigation, and Control Conference. pp. 3406–3414 (2006)
3. Chao, H., Gu, Y., Napolitano, M.: A survey of optical flow techniques for uav navigation applications. In: Unmanned Aircraft Systems (ICUAS), 2013 International Conference on. pp. 710–716. IEEE (2013)
4. Civera, J., Grasa, O.G., Davison, A.J., Montiel, J.M.M.: 1-point ransac for ekf-based structure from motion. In: International Conference on Intelligent Robots and Systems. pp. 3498–3504. IEEE (2009)
5. Ding, W., Wang, J., Han, S., Almagbile, A., Garratt, M.A., Lambert, A., Wang, J.J.: Adding optical flow into the gps/ins integration for uav navigation. In: Proc. of International Global Navigation Satellite Systems Society Symposium. pp. 1–13. Citeseer (2009)
6. Engels, C., Stewénius, H., Nistér, D.: Bundle adjustment rules. Photogrammetric computer vision 2, 124–131 (2006)
7. Griffiths, S., Saunders, J., Curtis, A., Barber, B., McLain, T., Beard, R.: Obstacle and terrain avoidance for miniature aerial vehicles. In: Advances in Unmanned Aerial Vehicles, pp. 213–244. Springer (2007)
8. Hrabar, S.: 3d path planning and stereo-based obstacle avoidance for rotorcraft uavs. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. pp. 807–814. IEEE (2008)
9. Martinez-Carranza, J., Calway, A.: Efficient ekf-based visual odometry using a structure-driven temporal map. In: International Conference on Robotics and Automation ICRA. IEEE (May 2012)
10. Martınez-Carranza, J., Loewen, N., Márquez, F., Garcıa, E.O., Mayol-Cuevas, W.: Towards autonomous flight of micro aerial vehicles using orb-slam. In: IEEE 3rd Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS (2015)
11. Mueller, T.J.: Fixed and flapping wing aerodynamics for micro air vehicle applications, vol. 195. AIAA (2001)
12. Mur-Artal, R., Montiel, J., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. Robotics, IEEE Transactions on 31(5), 1147–1163 (2015)
13. Roberts, J.M., Corke, P.I., Buskey, G.: Low-cost flight control system for a small autonomous helicopter. In: Australasian Conference on Robotics and Automation. Auckland, New Zealand (2002)

14. Scherer, S., Singh, S., Chamberlain, L., Saripalli, S.: Flying fast and low among obstacles. In: Robotics and Automation, 2007 IEEE International Conference on. pp. 2023–2029. IEEE (2007)
15. Shim, D., Chung, H., Kim, H.J., Sastry, S.: Autonomous exploration in unknown urban environments for unmanned aerial vehicles. In: Proc. AIAA GN&C Conference (2005)
16. Silveira Vidal, F., de Oliveira Palmerim Barcelos, A., Ferreira Rosa, P.F.: Slam solution based on particle filter with outliers filtering in dynamic environments. In: Industrial Electronics (ISIE), 2015 IEEE 24th International Symposium on. pp. 644–649. IEEE (2015)
17. Watanabe, Y., Calise, A.J., Johnson, E.N.: Vision-based obstacle avoidance for uavs. In: AIAA Guidance, Navigation and Control Conference and Exhibit. pp. 20–23 (2007)
18. Zhou, G., Fang, L., Tang, K., Zhang, H., Wang, K., Yang, K.: Guidance: A visual sensing platform for robotic applications. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 9–14 (2015)
19. Zufferey, J.C., Floreano, D.: Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. pp. 2594–2599. IEEE (2005)

Estudio comparativo de algoritmos de segmentación de piel usando atributos de color

Sheila Gonzalez-Reyna[1], Marlene Elizabeth López-Jiménez[1], Emmanuel Zavala-Mateo[1], Israel Yañez-Vargas[1], Jesús Guerrero-Turrubiates[2]

[1] Universidad Politécnica de Juventino Rosas, Salamanca, Gto., México

[2] Universidad de Guanajuato, División de Ingenierías Campus Irapuato-Salamanca, Salamanca, Gto., Mexico

{sgonzalez_ptc,marlene.lopez,emmanuel.zavala,jyanez_pa}@upjr.edu.mx
jdj.guerreroturrubiates@ugto.mx

Resumen. El presente artículo analiza una serie de métodos para el procesamiento de imágenes de personas, con el fin de segmentar su piel. La base de datos Jochen Triesch Static Hand Posture Database [9] proporciona una amplia gama de tonalidades de piel, obtenida de personas de razas y edades distintas. Las tareas de detección de piel en imágenes se enfrentan a diversos retos, debido a factores diversos como: variaciones en la iluminación, factores ambientales y fondo de la escena. En este trabajo se muestran los resultados obtenidos al desarrollar algoritmos de segmentación basados en distintas técnicas heurísticas y probabilísticas, a partir de la utilización de una base de datos en el espacio de color RGB.

Palabras clave: Segmentación de piel, umbralización, naive Bayes, perceptrón multicapa.

A Comparative Study of Skin Segmentation Algorithms Using Color Attributes

Abstract. This article analyzes a set of methods for processing images of people for skin segmentation. The Jochen Triesch Static Hand Posture Database [9] provides a wide range of skin tones obtained from people of different ages and breeds. The skin detection task on images give face to distinct challenges, due to various factors like: ilumination variations, environment factors and background. In this paper we show the results obtained in developing segmentation algorithms based on different heuristic and probabilistic techniques using RGB color space.

Keywords: Skin segmentation, thresholding, naive Bayes, multilayer perceptron.

1. Introducción

La detección de piel tiene un papel muy importante en una amplia gama de aplicaciones del procesamiento de imágenes, que van desde la detección y seguimiento de rostros, detección de gestos manuales y sistemas varios de interacción humano-computadora. Recientemente, métodos de detección de piel basados en información de color han sido ampliamente utilizados, debido a que la segmentación del color de piel es computacionalmente eficaz, característica deseable para la implementación de funciones más complejas.

Los algoritmos de segmentación permiten separar los objetos de interés del fondo o del resto de la escena en una imagen digital. La segmentación de imágenes es considerada una etapa importante en la detección y seguimiento de objetos. Algunas aplicaciones importantes incluyen reconocimiento de rostros y gestos manuales, interacción humano-computadora, entre otros.

La información de color en las imágenes digitales está representada en el espacio de color RGB (Red, Green and Blue). Sin embargo, este espacio de color se ve altamente afectado por cambios en la iluminación, dificultando las tareas de segmentación por color. Para sobrellevar este problema, existen distintos espacios de color, capaces de separar la intensidad del color, del color en sí, generando de esta manera, una cierta invarianza a los cambios en iluminación.

El presente artículo realiza un análisis de distintos métodos de segmentación de piel utilizando el color en espacio RGB como fuente de información, y realizando la tarea de segmentar por tres métodos distintos: umbralización, modelado del espacio de color mediante probabilidades y redes neuronales artificiales.

El resto del artículo se estructura de la siguiente manera. En la Sección 2 se presenta un resumen de algunos algoritmos relacionados. La Sección 3 describe los conceptos teóricos en que se sustentan los experimentos. Los resultados experimentales se analizan en la Sección 4. Finalmente, la Sección 5 da las conclusiones finales y presenta las perspectivas de trabajo futuro.

2. Antecedentes

La segmentación de objetos en imágenes digitales puede realizarse mediante atributos de color, forma y textura. Las características de color pueden considerarse las más simples debido a que no suelen considerar datos de posición, rotación y escalamiento. Sin embargo, el color de los objetos se ve seriamente afectado debido a factores ambientales tales como sol, sombra, lluvia y niebla. Debido a lo anterior, la segmentación de imágenes utilizando atributos de color se implementa utilizando modelos de color alternativos, donde el color se separa de la iluminación. Algunos de los espacios de color más utilizados para llevar a cabo la tarea de segmentación de piel son RGB [1, 10], YCbCr [8, 9], HSV, HSI, CieLab [8, 12].

En [12] los autores realizan un estudio comparativo de nueve modelos de crominancia, incluyendo CIELab y CIELuv, utilizados en la segmentación de piel para la detección de rostros. Los autores de [8] realizan un experimento de

comparación en la segmentación de color de piel utilizando dos modelos de color: CIELab y YCbCr, dicho experimento lleva a la conclusión de que el modelo de color CIELab produce mejores resultados debido a su capacidad para representar una mayor cantidad de colores.

Además se han realizado trabajos en donde la segmentación de piel se realiza en espacio de color RGB. En [1], se usó la información de color RGB para clasificar piel mediante un árbol difuso. El porcentaje de reconocimiento obtenido en su base de datos es del 94.1 %. Santos y Pedrini propusieron otro método para la segmentación de piel en espacio de color RGB, utilizando histogramas de color y apoyándose en mapas de salientes [10]. Sobieranski et al. [11] propuso la segmentación usando el mismo espacio de color, y clasificadores de distancia no lineal (Mahalanobis).

De acuerdo con el método de implementación, los algoritmos de segmentación por color pueden ser de tres tipos distintos [9]:

1. **Umbralización**. Este método implica un análisis de colores, con la finalidad de encontrar rangos de valores entre los cuales se encuentran los colores del objeto de interés,
2. **Modelos probabilísticos**. Se tiene una base de datos de colores, y se generará un modelo utilizando probabilidades que permitan segmentar el objeto de su fondo [10, 11], y
3. **Algoritmos de clasificación**. Los algoritmos clasificadores tienen su origen en distintas teorías, ellos pueden realizar una clasificación de objeto/fondo utilizando la información de color [1].

Los métodos de segmentación por color que involucran el uso de clasificadores, han tenido una especial atención en años recientes [7]. Esto debido a que los clasificadores se muestran robustos en la generalización de la información, es decir, se ven menos afectados por cambios de iluminación que los métodos de umbralización. Guerrero-Curieses et al. [3] realiza una comparación del desempeño de distintos clasificadores en tareas de segmentación, usando también información de color en diferentes espacios (CIELab, RGB normalizado, YCbCr).

3. Métodos

En esta Sección se enumeran los conceptos necesarios para la reproducción de los experimentos.

3.1. Método de umbralización

El color de piel en los humanos se determina por la cantidad de pigmento "melanina" en la piel en función de la absorción de la radiación ultravioleta del sol. Las personas con grandes cantidades de melanina tienen piel oscura, y las personas con una cantidad pequeña de esta tienen piel blanca [7].

El primer paso para la clasificación de color de piel es la elección del espacio de color en que se trabajará. El color RGB es el espacio predeterminado

para la mayoria de formatos de imagen. Cualquier otro espacio de color es una transformación del RGB.

Los colores de piel humana difieren entre un grupo de individuos por muy pequeño que éste sea, incluso difiere en las zonas del cuerpo de un mismo individuo, y bajo cierto nivel de iluminación. Uno de los métodos más utilizados y más sencillos es la definición de límites o rangos (umbrales) de color, donde mediante una decisión se hará la agrupación de componentes espaciales. Los valores de los pixeles de una imagen que se encuentran dentro de los umbrales calculados son considerados pixeles de piel, de manera que

$$y(c) = \begin{cases} 1 & thr_{low} \leq c \leq thr_{high} \\ 0 & \text{cualquier otro caso,} \end{cases} \tag{1}$$

donde thr_{low} y thr_{high} son los umbrales inferior y superior, respectivamente para el canal de color que está siendo evaluado.

3.2. Clasificador Naive Bayes

El objetivo de las técnicas de clasificación es asignar a cada elemento de un conjunto la pertenencia a una clase determinada. Las clases identifican a un conjunto de elementos que comparten cierta similitud en una serie de características. Las distintas clases definidas dependen de la aplicación. Por ejemplo, en este caso el objetivo es distinguir entre dos clases: las que son "piel" y las que son "no piel".

De acuerdo con [10], se puede construir un modelo Bayesiano para la clasificación de pixeles que representan piel/no piel. Considerando que c representa el color de un pixel, se tiene la siguiente relación:

$$P(piel|c) = \frac{P(c|piel)P(piel)}{P(c|piel)P(piel) + P(c|\neg piel)P(\neg piel)}. \tag{2}$$

La Eq. (2) representa la regla de Bayes para la toma de decisiones. Las probabilidades pueden calcularse mediante una distribución Gaussiana:

$$P(c|clase) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(c-\mu)^2}{2\sigma^2}, \tag{3}$$

donde μ es el valor del promedio de cada componente de color, y σ^2 es la varianza de la misma componente. La regla de Bayes puede ser simplificada suponiendo que cada característica (R, G, B) es independiente de las otras dos, generando de este modo el clasificador Naive Bayes [6].

3.3. Perceptrón Multicapa

El Perceptrón Multicapa (**MLP**) es el tipo más popular de redes neuronales [5]. Se ha aplicado con éxito en muchos problemas de reconocimiento de patrones

debido a su capacidad de aprender complejas relaciones entre entradas y salidas no lineales y la habilidad para generalizar los datos dados.

El perceptrón multicapa utilizado en este artículo tiene su estructura similar a la que se observa en la Fig. 1. A la entrada se encuentran los valores R, G y B del pixel a clasificar. A la entrada de la neurona 1, se calcula la suma ponderada de cada una de las entradas, más una señal de activación, como se muestra en 4.

$$N_{1,entrada} = \begin{bmatrix} R\ G\ B\ 1 \end{bmatrix} \begin{bmatrix} W_{R,1}\ W_{G,1}\ W_{B,1}\ b_1 \end{bmatrix}^T, \tag{4}$$

donde b_1 es la señal de activación, y los pesos $W_{c,1}, c \in \{R, G, B\}$ son las ponderaciones de los valores del pixel hacia la neurona 1. Una vez calculado el valor $N_{1,entrada}$, la salida de la neurona se obtiene aplicando la función sigmoidal a dicho valor:

$$N_1 = \frac{1}{1 + \exp\left(-N_{1,entrada}\right)}. \tag{5}$$

Operaciones similares se realizan con las neuronas N_2, N_3 y N_4. Las salidas de las neuronas 3 y 4, etiquetadas como S_3 y S_4 se comparan para tomar la desición final [2].

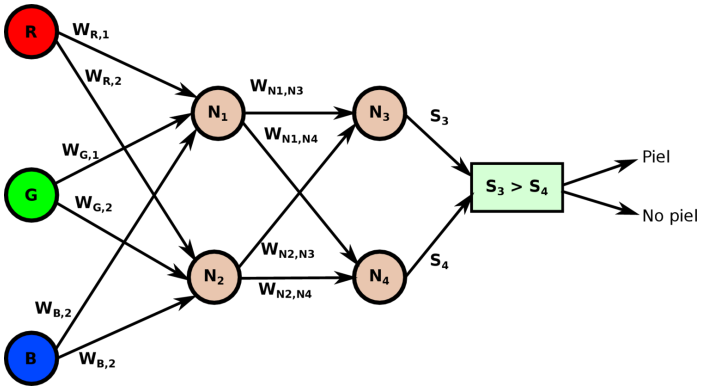


Fig. 1. Perceptrón multicapa para clasificación de color en espacio RGB

4. Resultados

Este artículo presenta un estudio comparativo de tres métodos para la segmentación de piel, que pueden ser utilizados para aplicaciones de interacción humano-computadora, detección de peatones, detección y reconocimiento de rostros, etc. Los métodos analizados son: 1) Umbralización, 2) clasificación usando Naive Bayes y 3) clasificación usando Redes Neuronales Multicapa (MLP). Estos métodos se aplican para lograr la segmentación de piel en imágenes en espacio de

color RGB ya que es la forma en la que se encuentra la base de datos; asimismo el tratamiento de imágenes en este espacio de color ahorra tiempo de procesamiento y muestra buenos resultados.

4.1. Base de datos de color de piel

Para la realización de los experimentos de este artículo, se utilizó la base de datos de Segmentación de Piel (Jochen Triesch Static Hand Posture Database) reportada en [1]. Esta base de datos está compuesta por un muestreo aleatorio de valores tomados de rostros de distintos grupos de edad (jóven, adulto, adulto mayor), grupos raciales (blanco, negro, asiático) y género. Está compuesta por un total de 200,045 muestras, de las cuales 50,859 pertenecen a muestras de tono de piel, y 149,186 son muestras de colores distintos a la piel. La Fig. 2 muestra las nubes de dispersión de la base de datos, graficada en dos canales de color de manera simultánea.

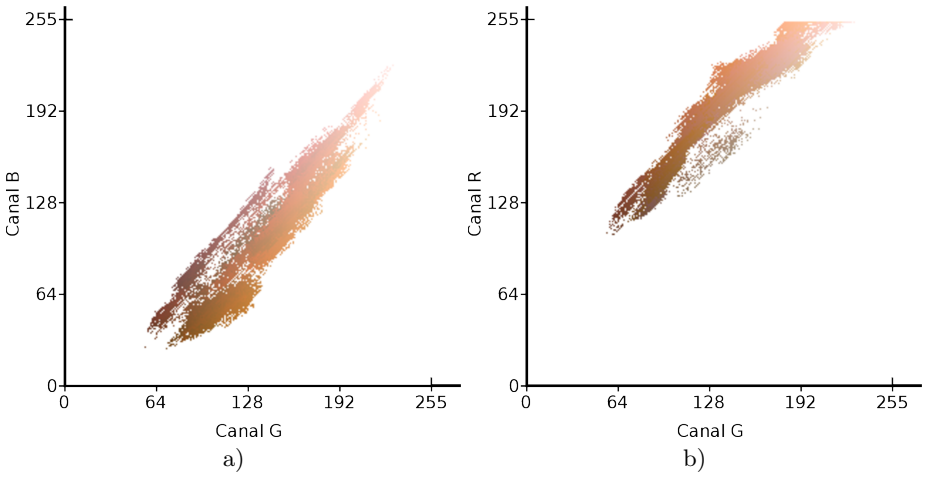


Fig. 2. Dispersión de la base de datos, a) canal G contra canal B, b) canal G contra canal R

4.2. Detección manual de umbrales para segmentación

La tarea de umbralización consiste en definir un rango de valores para cada canal de color, entre los cuales se encuentran los valores más comunes para el objeto de interés. La elección de los umbrales se realizó de forma heurística, es decir, acotando manualmente la nube de dispersión de la Fig. 2, en su región de mayor densidad, por inspección visual. Los valores de los umbrales escogidos se pueden observar en la Tabla 1.

Tabla 1. Umbrales propuestos para la segmentación, basados en la nube de dispersión de tonos de piel

Canal de color	Umbral menor	Umbral mayor
Azul	29	191
Verde	74	202
Rojo	130	250

Como se mencionó anteriormente, cada umbral es representado en el algoritmo de segmentación como una condición que controla el rango en cada canal(B,G,R) de la imagen sometida para cada pixel determinando así si este pertenece a la clase "Piel" en este caso con valor "1", o a la clase "No Piel" con valor "0" para así obtener como resultado una nueva imagen binarizada.

La Tabla 2 muestra la matriz de confusión del proceso de umbralización, que se obtuvo para los umbrales de la Tabla 1. Los valores "Verdaderos Positivos" y "Falsos Positivos", que se refieren a los datos que el algoritmo clasificó correctamente, son muy bajos en comparación con sus contrarios "Verdaderos Negativos" y "Falsos Negativos", respectivamente. Debido a lo anterior, se obtuvo un porcentaje bajo de eficacia (16.2028 %) en este proceso de segmentación.

Tabla 2. Matriz de confusión utilizando método de umbralización

Clase "Piel"	**Clase "No Piel"**	
7,690	43,169	**Clase "Piel"**
124,463	24,723	**Clase "No Piel"**

4.3. Segmentación utilizando algoritmo Naive Bayes

La clasificación supervisada se basa en la disponibilidad de datos de entrenamiento, de los que se conoce a qué clase pertenecen; extrayendo características de estos elementos es posible diseñar un algoritmo clasificador que asigne a muestras futuras una clase determinada en función de sus características.

Uno de los algoritmos más populares de clasificación es el conocido como Naive Bayes, que para este caso, genera un modelo Gaussiano para cada canal de color, pues asume que son independientes. La Tabla 3 muestra la matriz de confusión del clasificador naive Bayes que segmenta piel a partir del color de un pixel. Puede observar una mejora consistente en la eficacia de la clasificación, que es de un 92.36 %, comparado con el 16.2 % que se obtuvo con la umbralización.

Tabla 3. Matriz de confusión utilizando el clasificador Naive Bayes

Clase "Piel"	**Clase "No Piel"**	
38,956	1,903	**Clase "Piel"**
3,378	145,808	**Clase "No Piel"**

4.4. Segmentación utilizando Redes Neuronales Multicapa

La tarea de segmentación se llevó a cabo utilizando un MLP. Esta red neuronal fue modelada utilizando Weka [4], y posteriormente se implementó en MATLAB ®.

La imagen que sea sometida al algoritmo será dividida en los tres canales de color para su posterior procesamiento. La red diseñada consta de cuatro nodos distribuidos en dos capas. Los valores de entrada a la red son los tres canales de color RGB de la imagen sometida y como salida se obtiene una etiqueta para el caso de cada pixel determinando si estos pertenecen a la clase "piel" o "no piel". El resultado es una nueva imagen binarizada, a la cual corresponden valores de "0" y "1" para las clases "No piel" y "Piel" respectivamente. Esta nueva imagen tiene las mismas medidas que la imagen original pero en blanco y negro.

Tabla 4. Matriz de confusión utilizando clasificador MLP

Clase "Piel"	**Clase "No Piel"**	
50,597	262	**Clase "Piel"**
768	148,418	**Clase "No Piel"**

En los resultados obtenidos de la matriz de confusión para el MLP (Tabla 4) se puede observar que los "Verdaderos Positivos" y los "Falsos Positivos" son cifras muy grandes, lo cual quiere decir que el porcentaje de eficacia es muy alto al utilizar este algoritmo de segmentación (99.48 %).

4.5. Comparación de resultados obtenidos por los tres métodos

La Fig. 3.b) muestra el resultado de segmentar la imagen de la Fig. 3.a) mediante el proceso de umbralización. Observe que a pesar del bajo porcentaje de clasificación, la piel de las personas se segmenta correctamente. Por otro lado, hay información que debería formar parte del fondo (no es piel) y la umbralización no la descarta (falsos positivos).

La Fig. 3.c) despliega el resultado de la segmentación de la misma imagen, al utilizar el clasificador Naive Bayes para la toma de decisiones. Observe la mejora de los resultados, al existir una menor cantidad de falsos positivos (pixeles clasificados como piel, que realmente no lo eran).

Fig. 3. Resultados de la segmentación de una imagen. a) Imagen original. Segmentación utilizando b) umbralización, c) Naive Bayes y d) MLP

La imagen segmentada de la Fig. 3.d) se obtuvo mediante la red neuronal MLP. Observe una mayor limpieza en las zonas segmentadas, además de una cantidad considerablemente menor en el número de falsos positivos. Lo anterior se debe a los valores altos de sensitividad (98.5 %) y especificidad (99.82 %) obtenidos tras el proceso de clasificación.

De acuerdo con los datos de la Tabla 5, es posible verificar la superioridad de un algoritmo de clasificación en una tarea de segmentación, cuando se compara contra el resultado obtenido por la umbralización. Considerando la misma tabla, los porcentajes globales tienen cierta variación, teniendo un 16.20 % de eficacia en segmentación por "umbralización", y una gran diferencia en el porcentaje global del método por "Naive Bayes" obteniendo un 92.36 % y obteniendo un porcentaje de clasificación casi perfecta por el método "MLP" logrando un 99.48 % de eficacia, teniendo menos del 1 % como posible falla.

5. Conclusiones

En este proyecto se realizó el proceso de segmentación de color de piel, mediante tres algoritmos distintos para la base de datos Jochen Triesch Static

Tabla 5. Comparación en la eficiencia de los tres métodos de segmentación

Descripción	Umbralización	Naive Bayes	MLP
Porcentaje de pixeles clasificados correctamente	16.2028 %	92.3612 %	99.4851 %
Porcentaje de pixeles clasificados incorrectamente	83.7971 %	7.6388 %	0.5149 %
Sensitividad	5.8190 %	92.0205 %	98.5048 %
Especificidad	36.4151 %	92.456 %	99.8237 %

Hand Posture Database en espacio de color RGB. Se observó que el algoritmo basado en redes neuronales tiene mejores resultados para la detección de piel, sobre el algoritmo probabilístico de Naive Bayes y el método de umbralización, respectivamente. El espacio RGB crea muchos conflictos en el procesamiento dependiendo la iluminación del lugar donde sea tomada la imagen, sin embargo, los resultados mostrados aquí demuestran que dicho espacio es suficiente para llevar a cabo tareas de segmentación de piel, en ambientes con iluminación controlada (ambientes de interior, por ejemplo).

Este estudio se llevó a cabo para elegir el mejor proceso de segmentación y reducción de zonas de interés en imágenes que posteriormente serán sometidas a algoritmos de reconocimiento de gestos manuales. De la misma manera, las técnicas aquí exploradas pueden tener aplicaciones en otras tareas como detección de rostros o peatones.

Referencias

1. Bhatt, R.B., Sharma, G., Dhall, A., Chaudhury, S.: Efficient Skin Region Segmentation Using Low Complexity Fuzzy Decision Tree Model. In: 2009 Annual IEEE India Conference. pp. 1–4. Institute of Electrical & Electronics Engineers (IEEE) (2009), http://dx.doi.org/10.1109/INDCON.2009.5409447
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons (2012)
3. Guerrero-Curieses, A., Rojo-Álvarez, J.L., Conde-Pardo, P., Landesa-Vázquez, I., Ramos-López, J., Alba-Castro, J.L.: On the Performance of Kernel Methods for Skin Color Segmentation. EURASIP Journal on Advances in Signal Processing 2009(1), 1–13 (2009), http://dx.doi.org/10.1155/2009/856039
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software. SIGKDD Explor. Newsl. 11(1), 10–18 (nov 2009), http://dx.doi.org/10.1145/1656274.1656278
5. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall (1999), https://books.google.com.mx/books?id=3-1HPwAACAAJ
6. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338–345. UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995), http://dl.acm.org/citation.cfm?id=2074158.2074196

7. Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. Pattern Recognition 40(3), 1106–1122 (mar 2007), http://dx.doi.org/10.1016/j.patcog.2006.06.010
8. Kaur, A., Kranthi, B.: Comparison between YCbCr color space and CIELab color space for skin color segmentation. International Journal of Applied Information Systems, IJAIS 3(4), 30–33 (2012)
9. Phung, S.L., Chai, D., Bouzerdoum, A.: A universal and robust human skin color model using neural networks. In: IJCNN International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222). vol. 4, pp. 2844–2849. Institute of Electrical & Electronics Engineers (IEEE) (2001), http://dx.doi.org/10.1109/IJCNN.2001.938827
10. Santos, A., Pedrini, H.: Human Skin Segmentation Improved by Saliency Detection. In: Computer Analysis of Images and Patterns, pp. 146–157. Springer Science & Business Media (2015)
11. Sobieranski, A.C., Chiarella, V.F., Barreto-Alexandre, E., Linhares, R.T.F., Comunello, E., von Wangenheim, A.: Color Skin Segmentation Based on Non-linear Distance Metrics. In: Progress in Pattern Recognition Image Analysis, Computer Vision, and Applications, pp. 143–150. Springer Science & Business Media (2014)
12. Terrillon, J.C., Shirazi, M., Fukamachi, H., Akamatsu, S.: Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In: Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580). pp. 54–61. Institute of Electrical & Electronics Engineers (IEEE) (2000), http://dx.doi.org/10.1109/AFGR.2000.840612

Detección y seguimiento de palmas y puntas de los dedos en tiempo real basado en imágenes de profundidad para aplicaciones interactivas

Jonathan Robin Langford-Cervantes, Moises Alencastre-Miranda, Lourdes Munoz-Gomez, Octavio Navarro-Hinojosa, Gilberto Echeverria-Furio, Cristina Manrique-Juan, Mario Maqueo

Tecnologico de Monterrey, Campus Santa Fe, Ciudad de México, México

malencastre@itesm.mx

Resumen. Este artículo presenta un método para detectar las puntas de los dedos y palmas de la mano a partir de imágenes de profundidad. La detección de las puntas de los dedos y la palma se hace por medio de análisis morfológico de la región de la mano extraída, es decir, mediante la transformada de distancia y un análisis de un esqueleto inscrito en una imagen de profundidad. El método propuesto está pensado para usarse en aplicaciones interactivas, como videojuegos, y aplicaciones de Interacción Humano-Computadora. Desarrollamos un videojuego de escultura digital 3D que utiliza el método propuesto.

Palabras clave: Seguimiento de manos, clustering, análisis morfológico, aplicaciones interactivas.

Real-Time Palm and Fingertip Tracking Based on Depth Images for Interactive Applications

Abstract. This paper presents a method to detect the fingertips and palm from a human hand from depth images. The palm and fingertip detection is performed via morphological analysis of the extracted hand region, that is, distance transform and an analysis of an inscribed skeleton in the depth image. The proposed method is intended to be used in interactive applications, such as video games, or Human Computer Interaction applications. We developed a 3D digital sculpting video game that uses the proposed method.

Keywords: Hand tracking, point clustering, morphological analysis, interactive applications.

1. Introducción

Recientemente ha habido mucho énfasis en la investigación enfocada a la interacción humano-computadora (HCI, por Human-Computer Interaction en

inglés), proponiendo crear interfaces que utilizan directamente las habilidades naturales de comunicación y manipulación de los humanos. De las diferentes partes del cuerpo, las manos son las que son más efectivas para ese objetivo, debido a su destreza para comunicación y manipulación.

Previamente, se ha utilizado hardware especializado para realizar dicha tarea (e.g. sensores ópticos, guantes de datos). Aunque con ellos se pueden obtener mediciones precisas en tiempo real, son incómodos y de costos elevados. Por esto, los métodos basados en visión por computadora han sido la corriente principal en este campo, ya que son más baratos y permiten una experiencia de interacción más natural.

En interfaces basadas en visión por computadora, comúnmente se utiliza detección y seguimiento de manos para permitir al usuario interacciones como son el control del cursor, navegación en 3D, y reconocimiento de gestos dinámicos. Sin embargo, realizar el seguimiento de las manos correctamente es un reto, ya que los muchos grados de libertad en ellas implican una gran variabilidad en su apariencia, además de auto-oclusiones. Con la llegada de las cámaras de luz estructurada (también llamadas cámaras o sensores de profundidad, o RGB-D) y de tiempo de vuelo (time-of-flight), se ha vuelto disponible la información de profundidad en una imagen, lo que ha traído métodos pioneros para capturar la interacción del usuario.

Aunque hay muchas soluciones para seguimiento de manos basadas en cámaras de profundidad, presentan algunas limitaciones. Algunos enfoques dependen de métodos lentos de optimización, procesamiento en paralelo con tarjetas de video programables, o instalaciones inconvenientes [26]. Además, la mayoría no son adecuadas para el uso en aplicaciones interactivas, como videojuegos, debido a su complejidad computacional.

En el presente trabajo, se tiene como objetivo el desarrollo de un método que sea capaz de detectar las palmas y puntas de los dedos de una mano en base a la información obtenida con un sensor de profundidad. Se busca que la detección sea en tiempo real, con el propósito de que se pueda usar en aplicaciones interactivas, como son los videojuegos.

2. Trabajo relacionado

El seguimiento de palmas y puntas de dedos suele llevarse a cabo con métodos basados en visión monocular. Sin embargo, dada la flexibilidad de las manos, las auto-oclusiones, el color de la piel, y las condiciones de iluminación, entre otros, continúa siendo una tarea desafiante. Múltiples enfoques basados en visión para estimación de poses de manos y dedos han propuesto la integración de cámaras de profundidad, ya que son invariantes a la luz y al color de la piel.

Los métodos basados en modelos [12,17–19] usan un modelo virtual de mano que consiste en articulaciones cinemáticas, representando sus grados de libertad, para comparar con observaciones reales, y ajustarlas a las del modelo. En [17] se usan 8 cámaras sincronizadas. En cada una, mapas de color de piel y de bordes forman indicadores de la presencia de una mano, éstos se combinan y se

comparan con un modelo predefinido para deducir su pose. Esta implementación resuelve eficientemente un problema de seguimiento de manos con grados de libertad completos y oclusiones con el método de Particle Swarm Optimization, que puede paralelizarse eficientemente [25]. En GPU, se ejecuta en promedio a 15 cuadros por segundo, es sensible a cambios de iluminación, y requiere una instalación laboriosa.

Otro método basado en modelo se presenta en [26], utiliza tanto información de marcadores de un sistema óptico de captura de movimiento, como los datos RGB-D de un Kinect. Así, se adquiere un rango amplio y de alta fidelidad de datos de movimiento de las manos. Los dos dispositivos son complementarios entre sí, ya que se enfocan en diferentes aspectos de las acciones de las manos. Los sistemas basados en marcadores obtienen datos de posición 3D de alta resolución a gran velocidad, pero suelen no ser capaces de reconstruir precisamente las articulaciones de la mano en 3D, particularmente si hay auto-oclusiones. Complementado con la imagen RGB-D, el efecto se reduce significativamente, y se puede reconstruir la información del movimiento con alta calidad.

Si no se necesita detectar los grados de libertad de la mano, se puede utilizar un enfoque basado en características [6, 10, 13, 21] en el que sólo algunas, como las puntas de los dedos y el centro de la palma, o gestos simples, son detectados. Raheja et al. [21] utilizan un histograma de intensidad para detectar la dirección de la mano, el final de la muñeca, y el final de los dedos. Este método es rápido y eficiente para manos y puntas, pero sólo es útil cuando todos los dedos están presentes y abiertos completamente. Para tareas como señalar, cuando las puntas de los dedos son los elementos más importantes, Hongyong et al. [10] combinan información de color y profundidad para segmentar la mano y encontrar las puntas con detección de bordes. Los contornos aislados se tratan como manchas que se rastrean utilizando el algoritmo de clasificación KNN (K-Nearest Neighbors, o K-vecinos más cercanos), y después se interpretan como gestos.

Du et al. [6] segmentan la mano mediante umbralización de los datos de profundidad. Después, localizan los puntos convexos y cóncavos para predecir los dedos. Este método es capaz de detectar dedos en tiempo real con una precisión del 94 % para una mano. El trabajo presentado por Li et al. [13] también utiliza detección de convexidad, pero puede rastrear dos manos aplicando el algoritmo de K-Means después del umbral de profundidad. También puede detectar más gestos aplicando tres capas de clasificadores, con una precisión de 84 % para una mano, aumentando a más de 90 % si las dos hacen el mismo gesto. Liang et al. [14] obtuvieron errores de 0.69cm a 2.51cm en detección de dedos con un algoritmo que encuentra las palmas, y después restringe el movimiento posible a detectar, utiliza un rectángulo local, y puntos de cámino geodésico más corto para detectar las puntas de los dedos. Estos trabajos se enfocaron en obtener un alto porcentaje de precisión, pero no se ejecutan en tiempo real.

Suau et al. [23] lograron desempeño en tiempo real con una imagen de profundidad de baja resolución, siguiendo la cabeza y las manos relativas a la posición de ésta. Su desempeño en este aspecto fue mejor que en el SDK

del Kinect 1.0. El estado de la mano (abierto o cerrado) es detectado basado en el área delimitada por el conjunto de puntos. Lograron un error de menos de 10cm. Krejov et al. [11] usaron un detector Viola-Jones para encontrar la cara en una imagen de color, y con su posición establecieron un umbral para detección de manos. Después usaron el algoritmo de Dijkstra para recorrer un grafo ponderado construido a partir de los datos de profundidad del Kinect, para obtener 7 candidatos a puntas de dedo, y descartar falsos positivos como la muñeca. Sus resultados fueron un 80 % de detecciones entre 5.1 y 5.7mm de la posición real.

Hay aplicaciones como [5,9], en las que el problema de detección está resuelto, pero el método utilizado se explica sólo parcialmente. El segundo trabajo utiliza un Kinect para distinguir manos y dedos en una nube de más de 60,000 puntos a 30 cuadros por segundo. El primer trabajo presenta un seguidor de manos articuladas en tiempo real que reconstruye poses complejas con precisión, utilizando el Kinect del Xbox One. También se puede recuperar de fallos en el seguimiento.

El método propuesto busca obtener las puntas de los dedos y la palma de la mano, sin la necesidad de hardware especializado de gran costo. De igual manera, se busca que el método sea capaz de ejecutarse solamente con CPU, haciendo que sea viable su uso en computadoras portátiles, o de bajo costo. A diferencia de otros trabajos, se busca que el proceso de detección se logré en tiempo real, es decir, al menos poder procesar 30 cuadros por segundo. Se enfocó el trabajo en videojuegos, y aplicaciones interactivas similares, donde el uso de palmas y puntas de los dedos, así como gestos simples, como saludar, son suficientes para permitir control e interactividad. Para evaluar el método y proveer una forma de verificar la usabilidad de las manos en HCI, se desarrolló un videojuego de escultura digital 3D que utiliza el método tanto para modificar un modelo 3D preexistente, como para crear un objeto nuevo con "barro virtual".

3. Desarrollo

El método que se propone toma un mapa de profundidad como entrada desde un sensor Kinect, y trata de determinar la posición de las puntas de los dedos y las palmas de las manos para un sólo usuario. El proceso general es el siguiente:

1. Segmentación de las regiones potenciales de la mano usando el mapa de profundidad. El resto de los puntos del mapa se descartan.
2. Agrupamiento de puntos. Se usa un algoritmo de agrupamiento para juntar todos los puntos del mapa segmentado para posteriormente ser procesados.
3. Estimación de la posición de la palma utilizando la transformada de distancia.
4. Selección de puntas de los dedos. Se crea un grafo, que puede verse como el esqueleto para la mano, a partir de los grupos de puntos. Las ramas del grafo se escogen como candidatos a puntas de los dedos. Se aplican reglas adicionales para seleccionar las puntas finales.

3.1. Obtención del mapa de profundidad y segmentación

El mapa de profundidad debe ser segmentado para mantener solamente las regiones que corresponden a las manos y los antebrazos del usuario. La Figura 1 muestra un mapa de profundidad antes de ser filtrado, que comúnmente contiene elementos como la cabeza, los brazos, u otros objetos en escena. Para procesar solamente los puntos de manos y antebrazos, se creó una caja delimitadora que rodeará únicamente el área de interés, y deja fuera el resto de los puntos de la escena. Un mapa filtrado puede verse en la Figura 2. Escogimos este acercamiento para permitir a los usuarios de una aplicación configurar su espacio de trabajo y no restringir el posicionamiento del Kinect, o la gente usándolo.

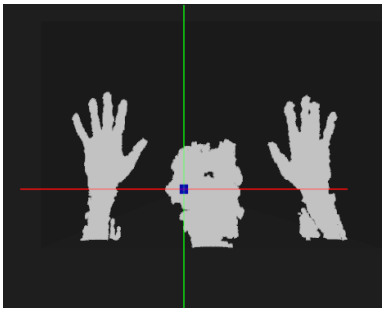


Fig. 1: Nube de puntos 3D antes de filtrarse

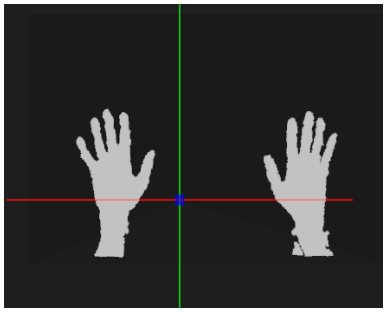


Fig. 2: Nube de puntos 3D después de filtrarse

3.2. Agrupamiento de puntos en grupos

Después de segmentar el mapa de profundidad, se agrupan los puntos 3D en grupos significativos que representan cada mano y antebrazo, para después obtener el grafo que servirá como esqueleto. Se consideró el uso de K-Means [15] y DBSCAN [8], debido a su eficiencia computacional y su efectividad al encontrar grupos significativos.

Se hicieron pruebas primero con el algoritmo K-Means, buscando en el mapa de profundidad filtrado dos grupos; uno para cada mano. El algoritmo de K-Means produjo dos grupos que se podían analizar posteriormente (como se ve en la Figura 3). Sin embargo, debido a que K-Means necesita un número predefinido de grupos a buscar, hubieron algunos casos en los que las manos estaban muy cerca entre sí, y ocasionaron que el algoritmo generara grupos con puntos que pertenecían a una mano asignados a la otra (ver Figura 4).

DBSCAN trabaja considerando la densidad de los grupos, y produce un número de grupos depende de los datos en sí, resolviendo así los problemas que se tienen con K-Means. Utilizando DBSCAN en un conjunto de datos de prueba, se encontró que los puntos que conforman las manos se agrupaban adecuadamente,

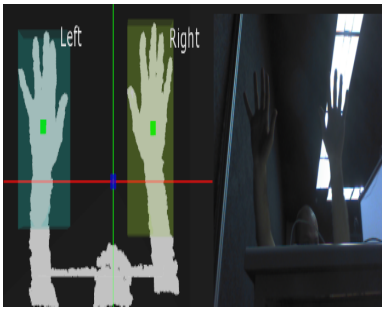


Fig. 3: Grupos generados correctamente usando K-Means

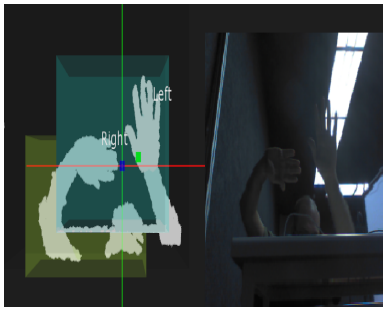


Fig. 4: Asignación incorrecta de puntos a un cluster con K-Means

por lo que se decidió seguir con dicho algoritmo. Sin embargo, el algoritmo no se ejecuta en tiempo real. Por esto, se utilizó una modificación a DBSCAN, propuesta por Navarro et al. [16], que es capaz de procesar nubes de puntos en tiempo real. La modificación a DBSCAN utiliza octrees y un esquema de particionamiento para reducir el tiempo que pasa buscando vecinos. El resultado del agrupamiento de la nube de puntos se puede ver en la Figura 5.

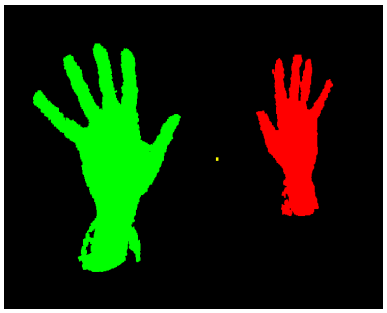


Fig. 5: Grupos finales obtenidos de la nube de puntos

3.3. Estimación de las palmas usando la transformada de distancia

Para estimar el centro de la palma, se utilizó el algoritmo de la transformada de distancia [20]. Para aplicar la transformada y estimar el centro de la palma, se siguieron los siguientes pasos:

- Se proyecta la nube ortogonalmente sobre el eje Z y se crea una imagen binarizada. El resultado se puede ver en la Figura 6 (a).
- Se rellena cualquier hoyo que se haya podido generar en el paso anterior. Mostrado en la Figura 6 (b).

- Se calcula la transformada de distancia de la imagen resultante. Se ve en la Figura 6 (c).
- Se encuentra el punto válido en el mapa de distancia con la distancia más grande. Si hay más de uno, se escoge el que tenga la menor coordenada Y. Ese punto servirá como el centro de la palma. Visto en la Figura 6 (d).

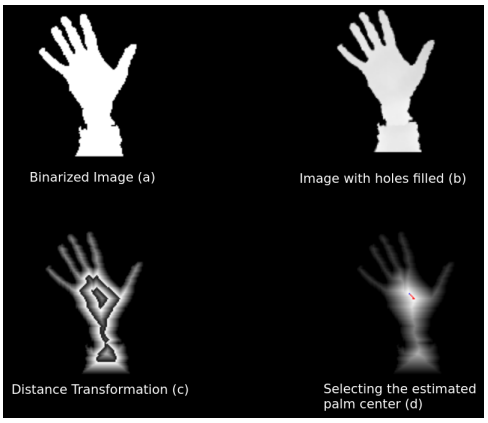


Fig. 6: Estimación del centro de la palma mediante la transformada de distancia en una imagen binarizada

3.4. Selección de puntas de los dedos

Para obtener las puntas de los dedos, se hace un análisis de los grupos obtenidos previamente para estimar la posición de las mismas. Se utilizó una técnica similar a la usada en [22, 24]. La idea principal es generar un esqueleto de curvas centradas de un grupo de puntos 3D, con el propósito de ayudar a estimar la forma de la mano y las puntas de los dedos.

Primero, se genera un octree con los puntos de cada grupo; esto permite utilizarlos en una búsqueda, y procesarlos en tiempo real. Para cada octree, se calcula el centroide de cada uno de sus voxeles, y se aplica DBSCAN para agruparlos a lo largo del eje Y, como se puede ver en la Figura 7. Para cada nuevo grupo que se genera, se calcula su centroide (ver Figura 8), de modo que se convierten en nodos de un esqueleto de curvas.

Para unir cada uno de los nuevos nodos, se diseñaron las siguientes reglas:

- Los nodos se conectarán de abajo hacia arriba en el eje Y.
- Cada nodo se conectará al siguiente conjunto de nodos más cercanos disponible.
- Los nodos buscarán conexiones solamente en los próximos cinco niveles del octree durante la búsqueda. Si el siguiente nodo más cercano está más lejos, no se conectará.

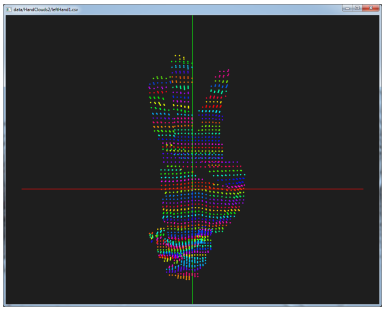


Fig. 7: Puntos del octree agrupados en grupos

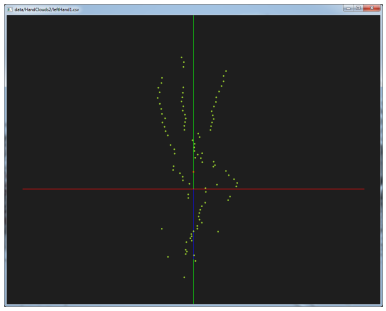


Fig. 8: Centroides de cada capa de los puntos en el octree

Usando esta técnica, se obtiene un grafo que puede verse como el esqueleto de curvas de cada mano. Se busca que los nodos finales del esqueleto sean los candidatos a puntas de los dedos. Sin embargo, el esqueleto generado puede tener múltiples bifurcaciones que deben de ser recortadas ya que también se considerarían como candidatos. Por esto, se eliminaron todos los caminos del esqueleto que tienen menos de dos conexiones subsecuentes.

Los nodos finales de cada camino son los que se considerarán como candidatos a puntas de dedos. Considerando la longitud de cada camino desde la raíz a los candidatos a dedo, todos los caminos se ordenan de forma descendiente, y sólo los primeros cinco se seleccionan. Para validar los candidatos, utilizamos un método geométrico. Definimos un radio de aprobación, que es inversamente proporcional al camino más largo desde el centro de la palma a los candidatos, y todos los que quedan fuera de ese radio se validan como puntas de dedo. En la Figura 9 se puede apreciar la selección de las puntas de los dedos.

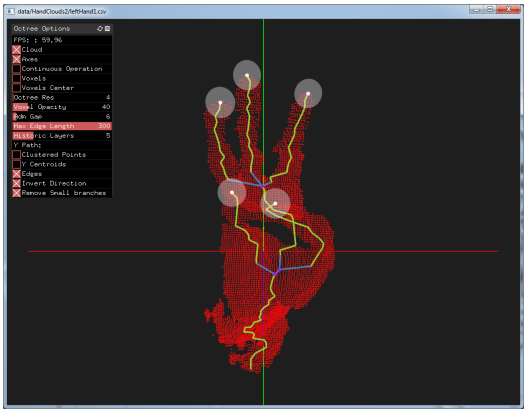


Fig. 9: Selección de puntas de dedos final

4. Resultados

El desarrollo se realizó en C++ utlilizando el SDK v1.8 del Kinect [1]. Se hicieron las siguientes pruebas con un conjunto de datos de muestra, y con una captura en vivo del Kinect. El conjunto de muestra consiste en quince diferentes mapas de profundidad, y cada uno consiste en alrededor de 30,000 puntos 3D, que capturan diferentes posiciones, tamaños, y número de manos. Tanto para el conjunto de muestra como para la captura en vivo, se fijó el Kinect en la misma posición. Se fijó a 80cm en frente del punto de captura, 40cm por debajo de éste, y con un ángulo de elevación de 25 grados.

4.1. Estimación de palmas y puntas de los dedos

La propuesta de solución es capaz de detectar la posición de las puntas de los dedos y las palmas de dos manos a un promedio de 60 cuadros por segundo. En las siguientes Figuras, las puntas de los dedos están marcadas con círculos verdes (pequeños), mientras que las palmas de las manos están marcadas con un círculo rojo (grandes). La Figura 10 muestra una captura en vivo de nuestro sistema, siguiendo palmas y puntas de los dedos de dos manos.

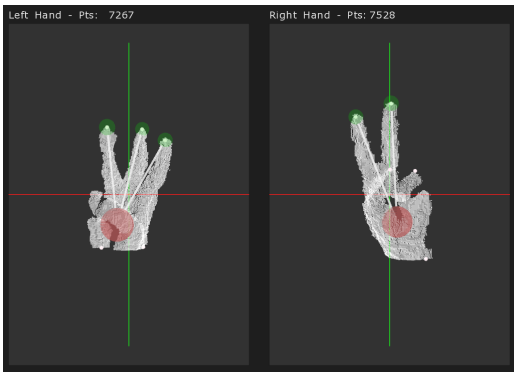


Fig. 10: Captura en vivo de la solución propuesta, siguiendo dos manos.

El sistema puede seguir las palmas y puntas detectadas mientras las manos estén apuntando hacia el Kinect, y las manos no estén ocluidas. El sistema también puede determinar si no hay ninguna punta de dedo presente, como se puede apreciar en la Figura 11 donde sólo el puño se muestra, y sólo la palma se sigue.

El radio de aprobación de la sección 3.4 ayudó con la detección para manos de distintos tamaños. Sin embargo, aunque este nos permitió filtrar algunos candidatos a dedos, también filtra el pulgar cuando está dentro del radio mencionado, como se ve en la Figura 12, donde una mano con 4 dedos apuntando hacia arriba se muestra.

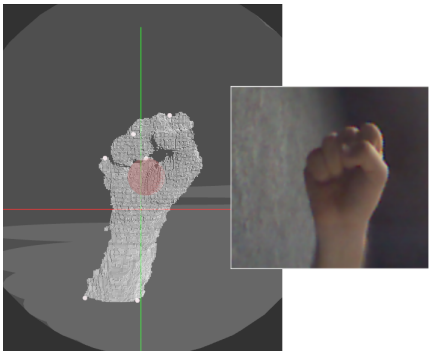


Fig. 11: Sólo la palma es seguida cuando se muestra un puño

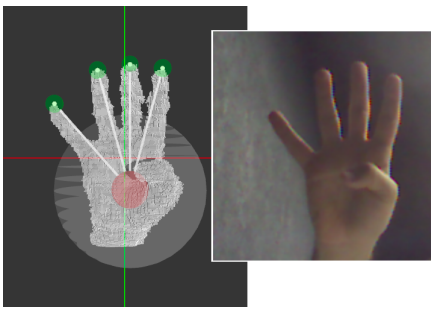


Fig. 12: Sólo 4 puntas de dedo y el centro de la palma se siguen, de una mano mostrando sólo 4 dedos

Para probar la solución más a fondo, se desarrolló un juego simple de escultura digital utilizando el motor de juegos Unity [7]. Dentro del juego, un conjunto de gestos (como agarrar y soltar) fueron programados utilizando la información obtenida con la detección y seguimiento de las palmas y puntas de los dedos. Con esos gestos, pudimos controlar la interfaz de usuario (UI) y los objetos en escena. La UI consistió en botones y barras deslizantes, con los que se interactúa abriendo y cerrando la mano, y arrastrando en el caso de las barras. El juego tiene dos modos: Escultura de barro digital, donde se puede modelar utilizando marching cubes imitando el barro, y escultura digital, donde se usan o deforman formas básicas, como una esfera o un toroide, con el objetivo de modelar algo distinto. Este juego nos mostró que la solución es lo suficientemente precisa como para desarrollar una aplicación sencilla de HCI, que permitió al usuario modelar precisamente diversos objetos. Una muestra del juego se puede ver en las Figuras 13 y 14.

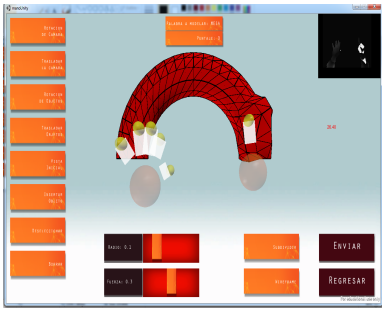


Fig. 13: Juego de ejemplo, esculpiendo un objeto nuevo utilizando un toroide como base

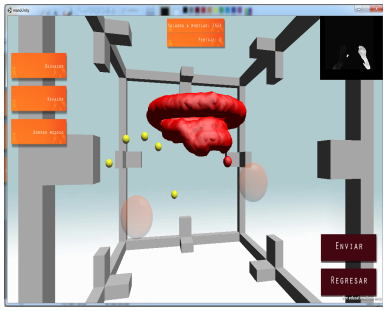


Fig. 14: Juego de ejemplo, modelando un objeto nuevo utilizando marching cubes como barro

5. Conclusiones y trabajo futuro

Se desarrolló un sistema que puede detectar la posición en 3D de las dos palmas y las puntas de los dedos de una mano humana, en tiempo real, utilizando un mapa de profundidad 3D. Aún cuando nuestra solución sólo detecta puntas de los dedos y centros de la palma, y puede perder el pulgar de las manos, produce un resultado aceptable para aplicaciones que necesitan una interfaz simple, y no tienen hardware dedicado, como tarjetas de video, disponible para procesamiento. Algunas de estas aplicaciones pueden ser videojuegos, simuladores, y con la llegada de los sensores de profundidad móviles [2–4], aplicaciones para dispositivos móviles.

Para trabajo a futuro, nos gustaría generar un modelo que detecte los dedos completos en cada mano. Además, la generación del esqueleto de curvas depende de la orientación de la mano (que necesita estar apuntando hacia arriba de frente al sensor de profundidad), por ello nos gustaría implementar un vector de orientación que pueda identificar la orientación de la mano, de modo que el requerimiento previo no sea necesario, generando así una mayor libertad dentro de los movimientos disponibles de las manos.

Referencias

1. Kinect for windows sdk v1.8 (sep 2014), http://www.microsoft.com/en-us/download/details.aspx?id=40278, Accesado: 2014-09-13
2. Atap project tango - google (sep 2014, Accesado: 2014-09-13), https://www.google.com/atap/projecttango/#project
3. Softkinetic - depthsense modules (sep 2014, Accesado: 2014-09-13), http://www.softkinetic.com/products/depthsensemodules.aspx
4. The structure sensor is the first 3d sensor for mobile devices (sep 2014, Accesado: 2014-09-13), http://structure.io/

5. Andrew, F., Daniel, F., Shahram, I., Cem, K., Eyal, K., Ido, L., Christoph, R., Toby, S., Jamie, S., Jonathan, T., Alon, V., Yichen, W.: Fully articulated hand tracking (2014), http://research.microsoft.com/en-us/projects/handpose/default.aspx, Accesado: 2014-10-29
6. Du, H., To, T.: Hand gesture recognition using kinect. Techical Report, Boston University (2011)
7. Engine, U.G.: Unity, http://unity3d.com/, Accesado: 2014-09-13
8. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining. vol. 96, pp. 226–231. AAAI Press, Portland, OR, USA (1996)
9. Garratt, G., Leslie, P.K., Russ, T., Tomas, L.P.: Kinect hand detection (2010), http://www.csail.mit.edu/videoarchive/research/hci/kinect-detection, Accesado: 2014-10-29
10. Hongyong, T., Youling, Y.: Finger tracking and gesture recognition with kinect. In: 2012 IEEE 12th International Conference on Computer and Information Technology (CIT). pp. 214–218. IEEE, Chengdu, Sichuan, China (2012)
11. Krejov, P., Bowden, R.: Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima. In: Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on. pp. 1–7. IEEE (2013)
12. Kuznetsova, A., Rosenhahn, B.: Hand pose estimation from a single rgb-d image. In: Advances in Visual Computing, Lecture Notes in Computer Science, vol. 8034, pp. 592–602. Springer Berlin Heidelberg (2013)
13. Li, Y.: Hand gesture recognition using kinect. In: 2012 IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS). pp. 196–199. IEEE, Haiddian District, Beijing, China (2012)
14. Liang, H., Yuan, J., Thalmann, D.: 3d fingertip and palm tracking in depth image sequences. In: Proceedings of the 20th ACM international conference on Multimedia. pp. 785–788. ACM (2012)
15. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. California, USA (1967)
16. Navarro-Hinojosa, O., Alencastre-Miranda, M.: Dbscan modificado con octrees para agrupar nubes de puntos en tiempo real. In: Memorias del 8º Congreso Mexicano de Inteligencia Artificial. INAOE (2016)
17. Oikonomidis, I., Kyriazis, N., Argyros, A.: Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: 2011 IEEE International Conference on Computer Vision (ICCV). pp. 2088–2095. IEEE, Barcelona, Spain (Nov 2011)
18. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3d tracking of hand articulations using kinect. In: 22nd British Machine Vision Conference. vol. 1, p. 3. Dundee, United Kingdom (2011)
19. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Tracking the articulated motion of two strongly interacting hands. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1862–1869. IEEE, Providence, USA (2012)
20. Raheja, J.L., Chaudhary, A., Singal, K.: Tracking of fingertips and centers of palm using kinect. In: 2011 Third International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM). pp. 248–252. IEEE, Langkawi, Malaysia (2011)

21. Raheja, J.L., Das, K., Chaudhary, A.: Fingertip detection: A fast method with natural hand. International Journal of Embedded Systems and Computer Engineering 3(2), pp. 85–89 (2011)
22. Sam, V., Kawata, H., Kanai, T.: A robust and centered curve skeleton extraction from 3d point cloud. Computer-Aided Design and Applications 9(6), pp. 869–879 (2012)
23. Suau, X., Ruiz-Hidalgo, J., Casas, J.R.: Real-time head and hand tracking based on 2.5 d data. Multimedia, IEEE Transactions on 14(3), 575–585 (2012)
24. Tagliasacchi, A., Zhang, H., Cohen-Or, D.: Curve skeleton extraction from incomplete point cloud. In: ACM SIGGRAPH 2009 Papers. SIGGRAPH '09, vol. 28, pp. 71:1–71:9. ACM, ACM, New York, NY, USA (2009)
25. Venter, G., Sobieszczanski-Sobieski, J.: Particle swarm optimization. American Institute of Aeronautics and Astronautics journal 41(8), pp. 1583–1589 (2003)
26. Zhao, W., Chai, J., Xu, Y.Q.: Combining marker-based mocap and rgb-d camera for acquiring high-fidelity hand motion data. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 33–42. Eurographics Association (2012)

Detección de texto en imágenes digitales como estrategia para mejorar la recuperación de imágenes por contenido

Manuel Mejía-Lavalle[1], Mathias Lux[2], Carlos Pérez[1], Alicia Martínez[1]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor, México

[2] Klagenfurt University, Klagenfurt am Wörthersee, Austria

{mlavalle, carlospl, amartinez}@cenidet.edu.mx, mlux@itec.aau.at

Resumen. Los avances en los algoritmos para la detección de texto están permitiendo detectar de mejor manera el texto que aparece en imágenes digitales de escenas naturales. En particular la Transformada del Ancho del Trazo (*Stroke Widht Transform*) ha mostrado buenos resultados. Sin embargo, en el área de Recuperación de Imágenes Basada en Contenido, en donde se manejan características globales para eficientar el proceso de búsqueda, en general no se está usando la información de alto nivel del texto. Típicamente se está empleando la textura o el color para detectar las regiones de texto. En este trabajo se investiga el impacto de detectar texto usando características globales en beneficio de la Recuperación de Imágenes Basada en Contenido y se propone una estrategia. De la experimentación realizada se observa que, usando nuestra estrategia, se logra una mayor precisión (15%) en la recuperación de imágenes digitales.

Palabras clave: Recuperación de imágenes basada en contenido, características globales, detección de texto, stroke widht transform.

Digital Images Text Detection as Strategy to Improve Content-Based Image Retrieval

Abstract. Recent research advances in text detection allow us for finding text regions in natural scenes rather accurately. Global features in Content-Based Image Retrieval,however, typically do not cover such a high level information. While characteristics of text regions may be reflected by texture or color properties, the respective pixels are not treated in a different way. In this work we investigate the impact of text detection on Content-Based Image Retrieval using global features. Detected text regions are preprocessed to allow for different treatment by feature extraction algorithms, and we show that our strategy, for certain domains, leads to a much higher precision (15%) in Content-Based Retrieval.

Keywords: Content-based image retrieval, global features, text detection, stroke width transform.

1. Introducción

La detección de texto en escenas naturales ha tenido avances significativos en los últimos 5 años. El principal objetivo de los algoritmos de detección de texto es localizar el texto que aparece inmerso en una imagen digital. Normalmente lo que sigue después de la localización del texto es el reconocimiento del texto, es decir, convertir la imagen de texto en caracteres de texto (OCR por sus siglas del inglés *Optical Character Recognition*).

Uno de los mejores algoritmos recientes para detección de texto en imágenes es el conocido como Transformada del Ancho del Trazo ó *Stroke Widht Transform* (SWT) [1], el cual tiene la habilidad de encontrar texto en fotografías digitales independientemente del idioma en que esté escrito el texto. Para nuestra investigación la parte de OCR no es relevante. Lo que nos interesa es la detección del texto en el contexto de la Recuperación de Imágenes Basada en Contenido (CBIR por sus siglas del inglés *Content-Based Image Retrieval*), en donde se requiere procesar-buscar, a gran velocidad y en una inmensa cantidad de imágenes digitales, la o las imágenes más parecidas a la imagen que el usuario presenta al sistema como muestra. De esta manera, más que saber qué dice el texto inmerso en la imagen digital, nuestro interés se centra en localizar la región de la imagen en donde aparece el texto: su tamaño, posición, color y textura, que nosotros asumimos (como hipótesis) que permitirían mejorar la precisión, a bajo nivel, para la recuperación basada en contenido.

Para lograr lo anterior, nosotros pre-procesamos la imagen para:

(i) Obtener una máscara de la región en donde hay texto, aplicando un color homogéneo,

(ii) Obtener una máscara de la región complementaria, es decir, en donde no hay texto.

Después de esto se hace la indexación, usando características globales comúnmente empleadas en los sistemas CBIR y que está probado que funcionan bien en imágenes digitales de escenas naturales. Para realizar la evaluación de nuestra estrategia propuesta, usamos la base de datos de imágenes *SIMPLIcity* [2]. También experimentamos con la base de imágenes *Street View Text* [3], para tener en total 11 diferentes categorías de imágenes. En el presente artículo actualizamos las referencias y ampliamos los experimentos previamente presentados por nosotros mismos en [4], validando y ratificando con más casos experimentales los beneficios que se obtienen empleando la estrategia que proponemos y que siguen vigentes desde entonces.

El resto de este trabajo está organizado de la siguiente manera: en la Sección 2 presentamos el trabajo relacionado al tema de la estrategia que proponemos; en la Sección 3 describimos la estrategia propuesta; la Sección 4 detalla los experimentos realizados y los resultados obtenidos y la Sección 5 concluye y discute el trabajo a realizar en el futuro inmediato.

2. Trabajo previo relacionado

Según [5] y [6] existen dos grupos principales de enfoques para la detección de texto en imágenes digitales fijas, los basados en:

(i) Textura y
(ii) Regiones.

Los algoritmos del primer grupo tratan la imagen a diferentes escalas y generalmente son costosas en términos de recursos computacionales de tiempo y memoria. Los algoritmos pertenecientes al segundo grupo se basan en las propiedades características a nivel *pixel*, como lo son un color constante, grupos de *pixeles*, etc. Nuestra investigación está más relacionada con el este segundo grupo.

En la literatura especializada podemos encontrar numerosos enfoques publicadas a lo largo de los años [5, 6, 7, 8, 9]. Por ejemplo, en [1] se propone el algoritmo SWT para la detección de texto. Ahí se emplea la idea de encontrar los trazos (*strokes*) que constituyen una letra. Primero se detectan los bordes y luego se trazan líneas a los bordes vecinos. Si se encuentra un borde y la dirección del gradiente coincide en este punto con la dirección gradiente del borde original y además la distancia permanece estable a lo largo del borde, se asume que los dos bordes son los límites de un trazo de un cierto grosor o anchura. Después que se tiene identificada una letra candidata, se procede a identificar grupos de letras que forman palabras y, sobre todo esto, finalmente se dibuja una caja que delimita el texto así localizado. Se ha reportado que SWT se desempeña bien en escenas naturales que contienen texto, con la interesante característica extra de ser independiente al lenguaje en que esté escrito el texto (lo cual por cierto no es el caso del método descrito en [10]). En [11] se propone un método para detectar texto con orientaciones más bien arbitrarias en escenas naturales consideradas complejas. El enfoque de los autores es similar al presentado en [1], con la diferencia de que cada *pixel* dentro de un trazo es asignado al haz de anchura del trazo. En este caso el proceso de identificación de una letra se basa en la consistencia de la anchura del trazo.

Algunos grupos de investigación han demostrado la utilidad de la detección de texto para mejorar los procesos de los sistemas CBIR. En [12] los autores proponen una nueva técnica para usar áreas de texto en imágenes que van a ser procesadas por un sistema CBIR. Básicamente emplean el algoritmo antes presentado en [13] y cuyos resultados tienen los mismos valores de las métricas de *presicion* y *recall* que los obtenidos usando SWT; ellos extraen el centro de la caja que confina a la o las letras detectadas, obteniendo además la escala y la orientación, para crear así un vector de características para que el sistema CBIR funcione mejor. En [14] otro sistema CBIR se propone y se usa la idea de identificar mediante la segmentación y búsqueda del componente (es decir, el caracter ó letra en cuestión), aplicando propiedades geométricas de los componentes detectados para crear así vectores de características.

En el presente trabajo nosotros empleamos SWT [1] para identificar regiones de texto. Con estas regiones ya localizadas, nos interesa investigar la influencia que se tendría al usar características globales, como las conocidas con el nombre de *Pyramid Histogram of Oriented Gradientes* (PHOG) [15], *Auto Color Correlogram* (ACC) [16], *Fuzzy Color and Texture Histogram* (FCTH) [17] y el *Joint Composite Descriptor* (JCD), que también es descrito en [17].

A continuación detallaremos la estrategia que nosotros proponemos y con la cual se desea mejorar el rendimiento de los sistemas tipo CBIR.

3. La estrategia propuesta

Nuestra estrategia usa cuatro características globales ampliamente conocidas en el área de los sistemas CBIR y que se sabe tienen buen desempeño de manera individual (en [18] el lector interesado puede consultar un estudio comparativo). Estas características globales son, como ya se había mencionado anteriormente: *Pyramid Histogram of Oriented Gradientes* (PHOG) [15], *Auto Color Correlogram* (ACC) [16], *Fuzzy Color and Texture Histogram* (FCTH) [17] y el *Joint Composite Descriptor* (JCD) [17].

Para extraer las características globales de las imágenes nosotros usamos LIRE [18], que es una librería *open-source* desarrollada explícitamente para evaluar sistemas CBIR. A su vez la librería LIRE está basada en una máquina de búsqueda robusta, veloz y bien conocida en el medio llamada *Lucene*. La principal aportación de LIRE consiste en contener una amplia gama de algoritmos de características globales, como PHOG, ACC, FCTH y JDC, entre otros muchos.

En particular nosotros implementamos SWT en lenguaje Java, usando métodos disponibles en LIRE, tales como *Canny Edge Detector* (CED) y el cálculo del gradiente de magnitud y dirección. Para la implementación seguimos lo mencionado por los autores iniciales del SWT [1]. El umbral para considerar válido un haz en un trazo lo fijamos en $\pi/2$; el tamaño mínimo de una letra lo situamos en los 10 *pixeles* y el máximo en 300 *pixeles*; la cantidad mínima para considerar que existe un texto son dos letras juntas que tengan el mismo color, con una distancia entre letras que no exceda tres veces el tamaño de la letra y cuyo grosor de trazo no sea el doble con respecto a otra(s) letra(s).

Para verificar que nuestra implementación de SWT estuviera arrojando resultados correctos, realizamos una inspección visual sobre las mismas imágenes usadas en este artículo. Nuestra implementación previamente validada de SWT ya forma parte de LIRE (https://code.google.com/p/lire/).

Para realizar la evaluación de nuestra estrategia usamos dos bases de imágenes. Lo anterior porque consideramos que es necesario probar SWT con imágenes que tienen bastantes regiones de texto y a la vez que han sido usadas para sistemas CBIR.

La base de imágenes *SIMPLIcity* [2] es bastante conocida y usada en el contexto de los sistemas CBIR. Por otro lado, la base de imágenes *Street View Text* [3] fue concebida pensando en experimentar y evaluar algoritmos especializados en la detección de textos. Con estas dos bases de imágenes creamos 11 categorías de imágenes. Luego seleccionamos las primeras 100 imágenes de *Street View Text* para mantener la consistencia con *SIMPLIcity*, cuya cada categoría está compuesta por 100 imágenes. Al final creamos y obtuvimos una base para nuestra experimentación con 1,100 imágenes. En la Figura 1 se muestran ejemplos de las imágenes con las regiones de texto enmascaradas o cubiertas por rectángulos en color negro. Por el contrario, en la Figura 2 se muestran imágenes de ejemplo en donde las regiones en donde no hay texto son cubiertas de color negro. Este pre-procesamiento de enmascaramiento se

realizó sólo sobre las imágenes en donde SWT detectó texto; aquellas imágenes en donde SWT no localizó texto alguno, se dejaron sin cambios.

Fig. 1. Imágenes de ejemplo donde el texto es enmascarado en negro por SWT

Para evaluar objetivamente la capacidad de recuperación de imágenes de nuestra estrategia usamos la métrica conocida como *P10* (*Precision-at-10*) cuya definición matemática es:

$$Pk = 1/k \sum r(Xn),$$

donde *X1, X2, X3 ...Xn* son los resultados según los ordena el método CBIR y *r(Xn) = 1 si Xn es relevante y 0 de otra manera*.

De esta manera cada imagen de cada categoría es considerada como una consulta o *query*; es decir, al final tendremos 1,100 consultas posibles. Para cada consulta verificamos si las primeras 10 imágenes (*k = 10*) recuperadas pertenecen a la categoría respectiva. La interpretación entonces de un *Pk = 1* equivale a un resultado (recuperación de imágenes) perfecto, mientras que *Pk = 0* indica que ninguna de las 10 primeras imágenes recuperadas pertenecen a la categoría que se mostró inicialmente como ejemplo.

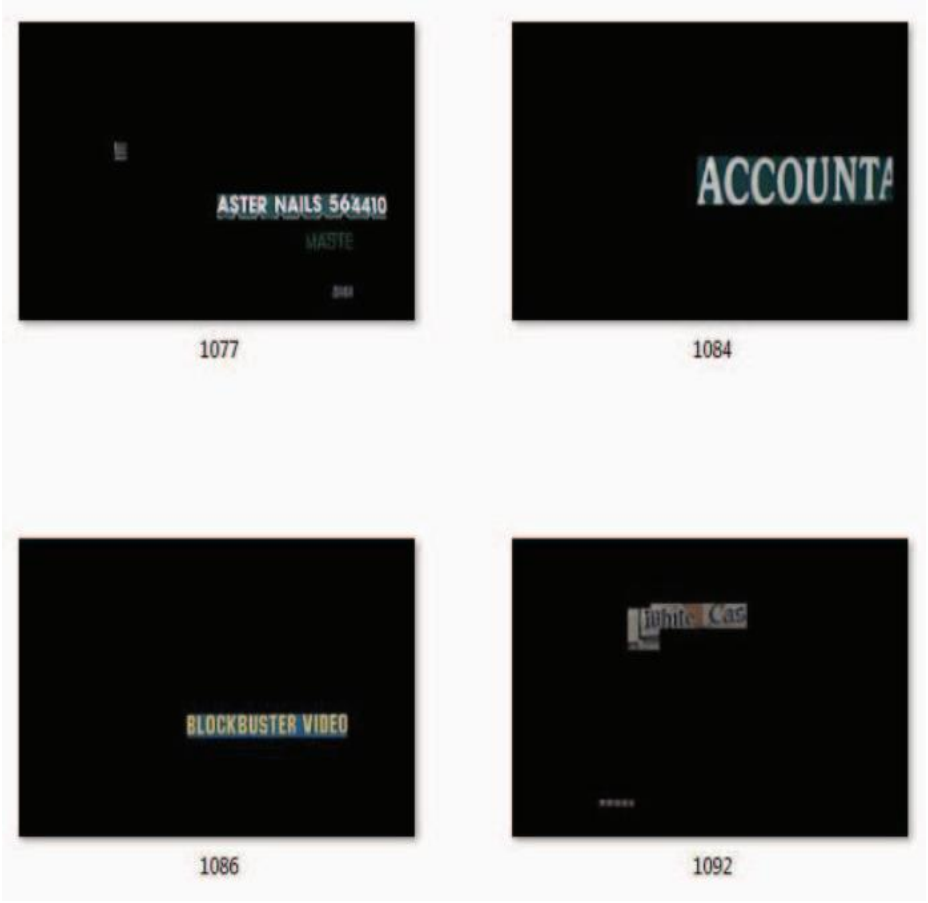


Fig. 2. Imágenes de ejemplo donde el texto no es enmascarado en negro por SWT, pero sí el resto de la imagen

4. Experimentación y resultados

En nuestra experimentación calculamos el valor *P10* para cada una de las 1,100 imágenes antes descritas y las promediamos por categoría. Puesto que nuestro objetivo inicial era investigar el impacto de la detección de texto cuando se usan las características globales (PHOG, ACC, FCTH y JDC) creamos dos conjuntos de imágenes de prueba: con el texto enmascarado en negro y con el resto de la imagen enmascarado en negro, dejando el texto visible (Figuras 1 y 2). Aún más, creamos tres versiones adicionales de prueba: las imágenes originales *D0*, las imágenes con el texto enmascarado *Dm* (Figura 1) y finalmente el complemento *Dt* (Figura 2). Las imágenes en donde SWT no detectó texto alguno, se dejaron sin aplicar ninguna máscara.

En las gráficas mostradas en las Figuras 3 a 5 las barras muestran el promedio de *P10* para cada una de las características globales de izquierda a derecha ACC, FCTH, JCD y PHOG respectivamente. La Figura 3 muestra los resultados sin detección de

texto *D0* y son los resultados de base para la comparación. La Figura 4 muestra el caso *Dm*, donde se observa una mejoría con respecto a *D0*; y la Figura 5 el caso *Dt*, donde se aprecian resultados no tan buenos como con *Dm*.

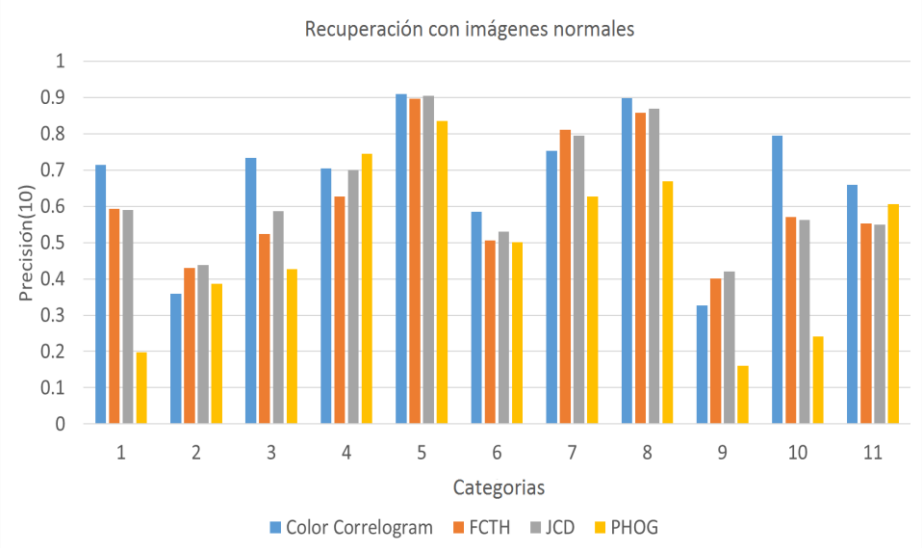


Fig. 3. Resultados cuando no se aplica pre-procesamiento alguno

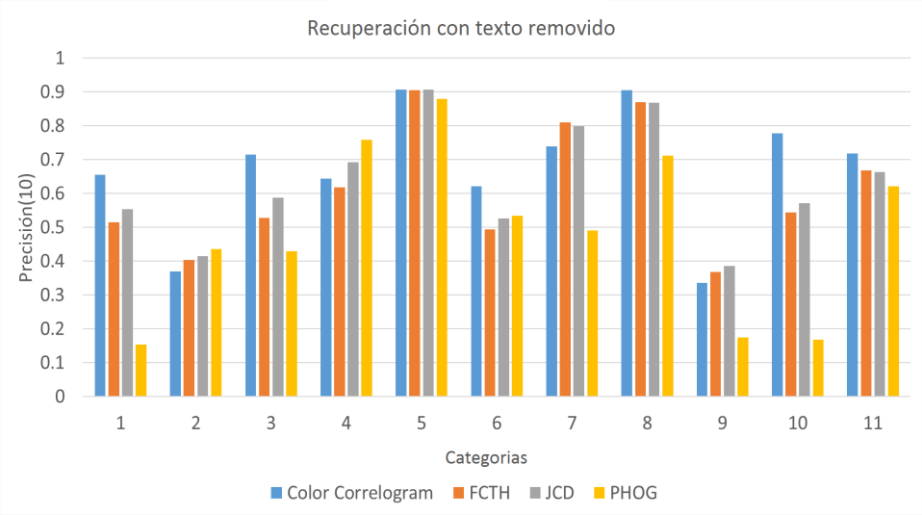


Fig. 4. Resultados cuando se aplica pre-procesamiento *Dm*

En la Tabla 1, a manera de resultados a detalle, se muestran los valores de *P10* de la categoría 11 usando las tres variantes de la base de imágenes. Se observa que cuando se incluye la detección de texto enmascarando el texto, se incrementa la precisión considerablemente para todos los cuatro extractores o descriptores de características globales investigados.

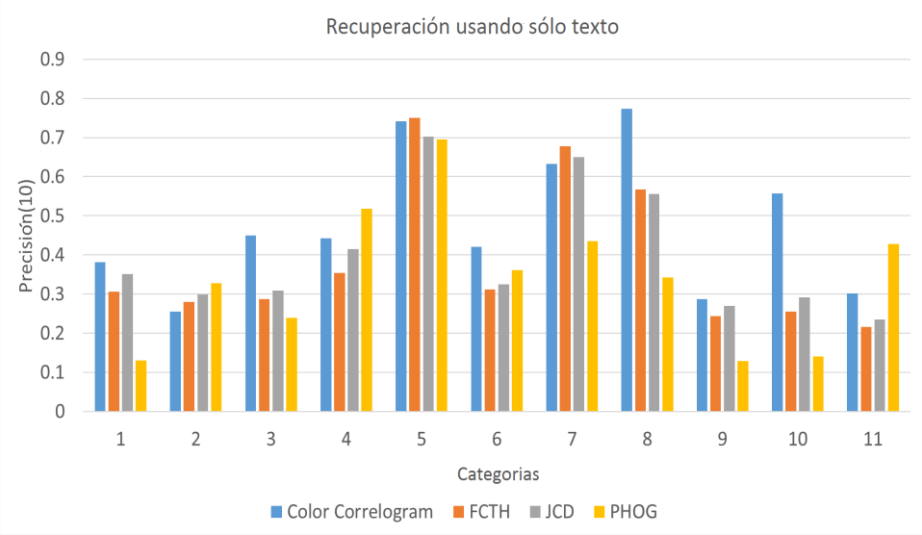


Fig. 5. Resultados cuando se aplica pre-procesamiento *Dt*

Tabla 1. Promedio de *P10* para la categoría 11, los mejores valores están en negritas

P10	**Variante**	**Descriptor**			
		ACC	JCD	PHOG	FCTH
Data Set	*D0*	0.658	0.550	0.605	0.552
	Dm	**0.717**	**0.664**	**0.621**	**0.668**
	Dt	0.302	0.235	0.428	0.215

En la Tabla 2 se muestra en porcentaje la mejoría promedio lograda por *Dm* comparada contra *D0*, obteniéndose al final una mejoría global de casi el 15%.

Tabla 2. Mejoría en porcentaje lograda por *Dm* en relación a *D0*

% mejor		**Descriptor**			
		ACC	JCD	PHOG	FCTH
Data Sets	*Dm* vs *D0*	+9	+21	+3	+21

5. Conclusiones y trabajo futuro

Presentamos una estrategia distinta para mejorar la recuperación de imágenes basada en contenido y que se beneficia de la detección de texto en las imágenes digitales por medio de SWT y cuatro descriptores globales comúnmente usados en los sistemas CBIR.

En nuestra experimentación calculamos el valor *P10* para cada una de las 1,100 imágenes de prueba y las promediamos por cada una de las 11 categorías que definimos.

Puesto que nuestro objetivo inicial era investigar el impacto de la detección de texto cuando se usan las características globales (PHOG, ACC, FCTH y JDC) creamos dos conjuntos de imágenes de prueba: con el texto enmascarado en negro y con el resto de la imagen enmascarado en negro, dejando el texto visible. Así creamos tres versiones adicionales de prueba: las imágenes originales *D0*, las imágenes con el texto enmascarado *Dm* y finalmente el complemento *Dt*.

Encontramos que la estrategia propuesta funciona bien (15% en promedio mejor) cuando el texto detectado es enmascarado en negro. En general, los trabajos del estado del arte revisados que buscan mejorar a los sistemas de recuperación de imágenes por contenido, logran mejorías por debajo de lo que nosotros logramos, observándose que aún mejorías inferiores al 10% ya son consideradas como más que aceptables por los autores de estos trabajos. Los resultados y mejoría alcanzada por nosotros creemos que significan una importante contribución al área, pues se pueden ver beneficiados dominios como la geo-localización, la búsqueda de instrumental especializado o la clasificación de escenas, por citar tan sólo tres casos en donde los sistemas CBIR son relevantes en el día a día.

Como trabajo futuro deberemos experimentar con más bases de imágenes y otros detectores globales. También se considera hacer pruebas con otros detectores de bordes, como se hizo en [19], que pudieran mejorar aún más los resultados reportados en el presente artículo. Algo interesante a experimentar es crear un algoritmo de fusión de los métodos basados en el procesamiento de la imagen y en combinación con la caracterización del texto detectado.

Referencias

1. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 2963–2970 (2010)
2. Wang, J.: *SIMPLIcity*: Semantics-sensitive Integrated Matching for Picture Libraries, 1 Introduction. Pattern Analysis and Machine Intelligence, Vol. 23, No. 9, pp. 947–963 (2001)
3. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: Computer Vision, IEEE International Conference on, pp. 1457–1464 (2011)
4. Perez, C., Lux, M., Mejia-Lavalle, M.: Toward Improving Content-Based Image Retrieval Systems by means of Text Detection. In: Mechatronics, Electronics and Automotive Engineering, IEEE Int. Conf on, pp. 50–53 (2014)
5. Gonzalez, A., Bergasa, L.M., Yebes, J.J., Bronte, S.: Text location in complex images. In: Pattern Recognition (ICPR), 21st International Conference on, IEEE (2012)
6. Li, Y., Lu H.: Scene text detection via stroke width. In: Pattern Recognition (ICPR), 21st International Conference on, IEEE (2012)
7. Lin, Z., Wu, Y., Zhao, Z., Fang, C.: A robust hybrid method for text detection in natural scenes by learning-based partial differential equations. Neurocomputing, Vol. 168, pp. 23–34 (2015)
8. Zhang, H., Zhao, K., Song, Y., Guo, J.: Text extraction from natural scene image: a survey. Neurocomputing, Vol. 122, pp. 310–323 (2013)
9. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. Vol. 37, No. 7, pp. 1480–1500 (2015)
10. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Computer Vision and Pattern Recognition (CVPR) IEEE Conference on, pp. 3538–3545 (2012)

11. Yi, C., Tian, Y.: Text string detection from natural scenes by structure-based partition and grouping. Image Processing, IEEE Transactions on, Vol. 20, No. 9, pp. 2594–2605 (2011)
12. Tsai, S., Chen, H., Chen, D., Parameswaran, V., Grzeszczuk, R., Girod, B.: Visual text features for image matching. In: Multimedia (ISM), IEEE International Symposium on, pp. 408–412 (2012)
13. Chen, H., Tsai, S.S., Schroth, G., Chen, D.M., Grzeszczuk, R., Girod, B.: Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In: Macqand, B., Schelkens, P. (eds), ICIP, pp. 2609–2612, IEEE (2011)
14. Nigam, A., Garg, A.K., Tripathi, R.: Content based trademark retrieval by integrating shape with colour and texture information. International Journal of Computer Applications, Vol. 22, No. 7, pp. 40–45 (2011)
15. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM, International Conference on Image and Video Retrieval, CIVR '07, pp. 401–408, New York, NY, USA ACM (2007)
16. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), Washington, DC, USA IEEE Computer Society (1997)
17. Chatzichristofis, S., Boutalis, Y.: Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In: Gasteratos, A., Vincze, M., Tsotsos, J. (eds), Computer Vision Systems, Vol. 5008 of Lecture Notes in Computer Science, pp. 312–322, Springer Berlin Heidelberg (2008)
18. Lux, M.: Lire: Open source image retrieval in Java. In: ACM International Conference on Multimedia (2013)
19. Mosleh, A., Bouguila, N., Hamza, A.B.: Image text detection using a bandlet-based edge detector and stroke width transform. In: Proceedings of the British Machine Vision Conference, pp. 63.1–63.12.BMVA Press (2012)

Combinación de un controlador PID y el sistema Vicon para micro-vehículos aéreos

Roberto Munguía, Aldrich Cabrera, Oyuki Rojas, José Martínez-Carranza

[1] Instituto Tecnólogico Superior de Atlixco,
México

[2] Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla,
México

direccion@itsatlixco.edu.mx, carranza@inaop.mx

Resumen. El sistema Vicon es un sistema de captura de movimiento que permite estimar la posición, en forma de traslación y orientación, de un cuerpo en movimiento. El sistema Vicon se conforma de un conjunto de cámaras que se colocan alrededor de un área de trabajo en la cual, el cuerpo a seguir por el sistema se desplaza al interior de dicha área. De este modo, la posición y orientación del cuerpo de interés, son estimadas a través del sistema Vicon. Por lo anterior, en este trabajo presentamos la implementación de un controlador Proporcional-Integral-Derivativo (PID), siendo este uno de los controles más usados en los sistemas robóticos, para que un micro-vehículo aéreo no tripulado pueda ejecutar vuelo autónomo entre dos puntos. El control PID implementado en este trabajo utiliza la posición y orientación del vehículo proporcionada por el sistema Vicon a una tasa de 300 cuadros por segundo. Adicionalmente, el sistema Vicon que se utilizó en este trabajo puede operar en ambientes exteriores, por lo que el conjunto de resultados que aquí se reportan incluyen experimentos de vuelo autónomo en exteriores.

Palabras clave: Control, vuelo autónomo, VANTS, drones, Vicon.

On Combining a PID Controller and the Vicon System for Micro-Aerial Vehicles

Abstract. Vicon is a motion capture system that enables the estimation of translation and rotation of a moving object. The Vicon system is a set of cameras placed around a work area, the object to track moves within such area. Therefore, position and orientation of the object of interest can be estimated through the Vicon system. Motivated by this, in this work we present the implementation of a Proportional-Integral-Derivative (PID), Since this is one of the most used controllers in robotics, thus aiming at enabling a micro-aerial vehicle to perform autonomous flight between two points. The PID controller implemented in this work utilizes the position and orientation estimates from Vicon,

which runs at a frequency of 300 Hz. In addition, the Vicon system used in this work can also operate in outdoors. From the latter, our experiments include autonomous flight runs in outdoors.

Keywords: Control, autonomous flight, UAVs, drones, Vicon.

1. Introducción

Un sistema de localización es una herramienta que permite monitorear con una alta precisión la ubicación de un objeto. Actualmente los vehículos aéreos no tripulados cuentan con Acelerómetros, giróscopos y GPS. Los cuales se pueden utilizar como herramienta de localización. Sin embargo tienen limitantes, una de ellas es el deficiente funcionamiento en interiores o incluso la perdida de comunicación, lo que evita controlar los movimientos bruscos que el vehículo aéreo pueda realizar durante la navegación.

Hoy en día, se implementan sistemas de localización en robots para la obtención de posición, para mejorar su control. Uno de los sistemas de localización más populares es el sistema Vicon [1], el cual proporciona una alta precisión en la medición de posición y el seguimiento de cuerpos rígidos utilizando cámaras infrarrojas. Al emplear un sistema Vicon como herramienta de localización externa e implementar un control PID en el vehículo aéreo, se mejora significativamente la navegación del vehículo al seguir una trayectoria entre dos puntos. Puesto que se reducen los movimientos bruscos que realiza durante la rutina.

Nosotros asumimos que el sistema Vicon resulta ser una mejor opción para capturar el movimiento del vehículo aéreo, ya que nos otorga datos concretos de la localización del objeto. Esto último se envía a nuestro algoritmo de control con el fin de calcular las señales de control, que serán enviadas la computadora y posteriormente al vehículo aéreo. Los resultados preliminares indicaran que incluso con la perdida de la trayectoria que puede tener nuestro vehículo aéreo llegaran los datos a nuestro controlador, el cual realizara los cambios necesarios para que el vehículo aéreo sea capaz de seguir la trayectoria con el objetivo definido.

Por lo tanto, a fin de presentar los resultados, este trabajo se organiza de la siguiente manera: la sección 2 presenta los trabajos relacionados; sección 3 describe nuestra configuración del sistema a preparar; La sección 4 describe nuestros experimentos realizados con el controlador y resultados donde se describen ampliamente lo ocurrido al momento del vuelo autónomo y finalmente en la sección 5 se presentan las conclusiones.

2. Trabajos relacionados

Los trabajos más relevantes en este tipo de investigación, especifica que debe existir una cantidad de parámetros que se deben tomar en cuenta al momento

de que el vehículo aéreo entre en vuelo autónomo y gracias a los algoritmos de reconocimiento de un checkpoit marcado de un color se puede aplicar un control PID para corregir los errores que presenta el cabeceo y balanceo [2].

Sin embargo, al momento de mover un robot en una posición arbitraria cuyo destino es una línea recta que el robot no conoce como tal, se recurre a lectura de sensores y encoders con alta resolución capaz de calcular la posición y la velocidad angular de éste para posteriormente implementarlo a un control PID compensando las desorientaciones que presente el robot [3].

Algunos sistemas robóticos poseen un control dentro de sus sistemas lo cual es un apoyo para que el robot realice ciertas tareas. Se conoce que en el área de la robótica los sistemas deben poseer más de un control dentro de sus sistemas, para posibles saturaciones propias en etapas de control [4]. Un sistema similar al de los robots móviles son los tipo robots SCARA puesto que los errores que poseen estos son en sus articulaciones al momento de orientarse y trasladar objetos en un circuito circular que puede presentar señales turbias al momento de realizar una tarea. Los trabajos sobre robots móviles deben considerar como una posible solución un control PID o de lógica difusa que pueda permitir correcciones al momento de la orientación y trayectoria que realicen durante una rutina de movimiento o de un vuelo no tripulado [5] [6].

Entre la comunidad científica se encuentran trabajos que requiere que los vehículos aéreos permanezcan estables y equilibrados en vuelo. Eliminando los efectos de las perturbaciones que afectan a los helicópteros y Quadrotors. Uno de ellos añade una carga extra al vehículo aéreo puesto que debe transportar objetos. Éste trabajo emplea un control que debe considerar los parámetros que ocasionan una desorientación, mediante tres controles PID (proporcional-integral-derivativo) [7].

Otro trabajo de interés con lo cual destaca por la utilización de la Validación de la Unidad de Medición Inercial (IMU) para la orientación del vehículo aéreo ofreciendo una robustez del control automático del cuadracóptero mediante señales de ruido que los IMU otorgan. Es importante debido a que el control PID requiere de señales de entrada para un correcto funcionamiento del sistema [8].

En algunos casos utilizan software de simulación como MATLAB o LABVIEW, ya que ayudan a leer datos de orientación como yaw, pitch y roll para posteriormente realizar la implementación de un control PID [9]. El controlador PID también puede ser implementado en sus motores con un sistema PWM, esto ayuda a corregir la inclinación manteniéndolo a un ángulo cercano a 0^o [10].

En la tesis de David Melero [11] se utilizó un PID para control de la orientación de un micro helicóptero donde utiliza varias herramientas como pasos para diseñar el controlador, mediante ayuda de software como MATLAB obtuvieron los datos para pasarlos a un filtro donde la estabilidad sea mucho mayor y finalmente con ayuda de IMU simulan los valores de posición y orientación monitorean asi al vehículo aéreo.

Tomando la misma idea del trabajo anterior, el autor de tesis Antonio Pico realizó un trabajo parecido, realizando su PID para cada uno de los parámetros de navegación, el autor realiza este controlador conectando IMU'S a una compu-

tadora *gunstix* implementada dentro de el vehículo aéreo. Sin embargo, comenta que para lograr un vuelo estable con apoyo del PID es necesario tener en cuenta cada [12].

Con los trabajos mencionados anteriormente se puede notar claramente que la diferencia de nuestro trabajo es la implementación del sistema vicon como herramienta de localización externa. Puesto que la mayoría usa sistemas IMU'S como herramienta de localización. Incluso utilizan cámaras de video que ayudaron a la creación de dichos trabajos.

3. Configuración del sistema

El proceso se realiza fuera del vehículo aéreo, en una computadora portátil con las siguientes especificaciones: procesador Intel Core i5 con 6 GB de RAM. El sistema de control se divide en dos etapas principales, las cuales funcionan dentro del sistema ROS. La primera etapa consiste en obtener la posición y orientación del vehículo aéreo mediante un sistema de localización Vicon. Los datos obtenidos por el sistema Vicon sirven posteriormente como entradas para el control PID, el cual es la segunda etapa. Las salidas del control PID son usadas para corregir la navegación del vehículo aéreo.

3.1. Control PID

El control PID (Proporcional, Integral, Derivativo) es un sistema que suministra entradas de control que son proporcionales a la diferencia entre las salidas actuales del sistema del vehículo aéreo y los valores de referencia del sistema Vicon. En la siguiente ecuación se muestra de manera generalizada los controladores PID en su forma matemática (1). Asimismo en la Fig. 1 se muestra en un diagrama a bloques el control PID, la planta (vehículo aéreo) y la retroalimentación (sistema Vicon). Introduciendo una retroalimentación se compensan perturbaciones como: desequilibrio y desorientaciones.

$$u(t) = Kp\left[e(t) + Ki\int_0^t e(t)dt + Kd\frac{de(t)}{dt}\right], \tag{1}$$

donde:

*t: Corresponde al tiempo transcurrido, o tiempo instantáneo.
*u (t): Corresponde a la salida del controlador.
*e (t): Corresponde al error (valor deseado-salida real del sistema).
*kp: Corresponde a la ganancia proporcional del controlador.
*ki: Corresponde a la ganancia integral del controlador.
*kd: Corresponde a la ganancia derivativa del controlador.

Se emplearon dos controles PID uno para corregir rotación del vehículo (*yaw*) y el segundo para corregir la traslación del vehículo (*pitch*), puesto que sólo se requería que el vehículo aéreo se desplazara en el plano *x,y*. Debido a que no se tiene un modelo matemático del vehículo aéreo, no se emplearon métodos analíticos para la obtención de ganancias. Por lo que las ganancias Kp, Ki y Kd

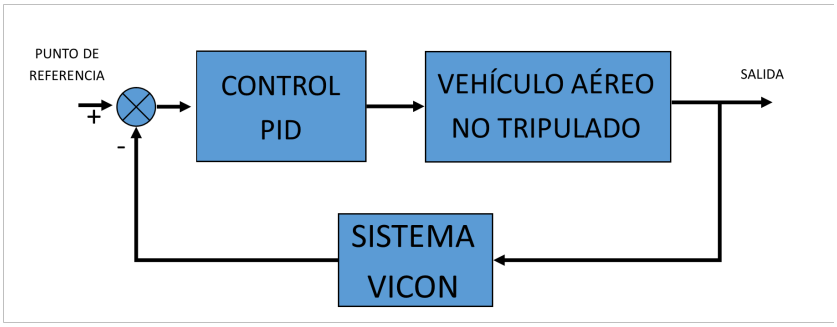


Fig. 1. Diagrama a bloques del control PID

de cada controlador PID se obtuvieron mediante experimentación, hasta obtener mejores resultados.

3.2. Sistema de localización Vicon

El sistema de localización Vicon es una solución al seguimiento de objetos de gran alcance, ya que proporciona una precisión de datos sin igual para la integración en aplicaciones 3D. El tipo de rastreo puede procesar datos en tan sólo 1,5 ms. a más de 500 cuadros por segundo, cinco veces más rápido que otros sistemas. Con la capacidad de reconocer los cuerpos rígidos en 2D, los datos continuarán incluso si los marcadores se hacen visibles a una sola cámara [1].

Este sistema de localización nos ofrecerá una experiencia de realidad virtual que se puede crear al momento de montar las cámaras en un área donde sus límites de enfoque no sean obstruidas por objetos dentro del sitio donde se localizará el vehículo aéreo, esta área debe contener al menos 3 cámaras que puedan seguir al vehículo en todo momento, generando así 3 parámetros de posición, x, y, z, así como también 3 parámetros de orientación, yaw, pitch, roll, de los cuales estos datos son almacenados en un archivo de texto que nos servirán como una herramienta para la creación del control PID al momento de vuelo autónomo.

4. Experimentos

Se realizaron dos tipos de experimentos en condiciones distintas, ambos con el objetivo de demostrar el vuelo autónomo en línea recta variando el ángulo inicial del vehículo con respecto al punto final. La primera prueba se realizó en interiores en un ambiente controlado, colocando en el vehículo aéreo una carcasa para prevenir colisiones y así proteger las hélices (Figura 2a). La segunda prueba se realizó en exteriores, en el cual no se incluyó en el vehículo la carcasa protectora, debido a que al usarla generaría arrastre a causa del viento y perturbaría en mayor medida la trayectoria seguida por el vehículo aéreo (Figura 2b).

Para ambas pruebas se generó un sistema de coordenadas de referencia por medio del software *"tracker 2.2 — VICON"*, donde es posible visualizar los ejes x, y, z, nos obstante el eje z no fue necesario para el control puesto que al vehículo aéreo se le definió una altura de 70cm en interiores y 1.5m en exteriores, debido a que se requería probar la efectividad del control PID en navegación a baja altura. El control PID comienza a ejecutarse después del despegue del vehículo aéreo. Éste al ubicarse sobre el punto final o al cruzar a -x, aterriza y el registro de los datos de la ubicación obtenidos por el sistema Vicon se detiene. Una vez obtenidas las señales de control se normalizan para que el valor se encuentre dentro del rango de -1 a 1, dado que es el rango permitido para controlar al vehículo aéreo.

(a) Vehículo Aéreo para Interiores

(b) Vehículo Aéreo para Exteriores

Fig. 2. Muestra de los vehículos aéreos preparados para los experimentos en el interior y exterior del laboratorio de robótica

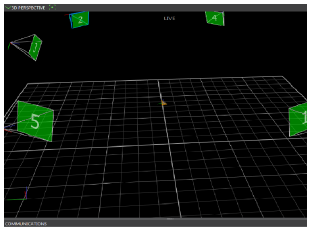
4.1. Pruebas en interiores

Se realizaron pruebas de vuelo del vehículo aéreo dentro del laboratorio, fue necesario conocer los límites en cuanto a maniobrabilidad y desplazamiento que se tenían al realizar el vuelo en interiores. Se estableció un área dentro del laboratorio donde se colocaron las cámaras Vicon, una vez montadas al rededor del área, se calibraron para obtener mayor precisión de la localización del vehículo aéreo. Para esta prueba se realizaron 10 experimentos en los que se utilizó el punto (0,0) como referencia para así calcular las señales de control para pitch y yaw del vehículo aéreo. Dichas pruebas consistieron en lo siguiente: el vehículo aéreo se posicionó a 2.5m sobre el eje x, a una altura de 70cm, con una orientación de 90^o respecto del punto final. Éste debía llegar hasta un punto previamente determinado, en este caso el punto (0,0).

En la Fig. 4a se muestra una vista superior de la trayectoria seguida por el vehículo aéreo al ser pilotado por el control PID, nótese que el punto de aterrizaje en los 10 experimentos realizados se encuentra en promedio a 2.32cm del punto (0,0), con una desviación estándar de 0.82cm. Con estos datos se pudo notar que nuestro controlador posee precisión.

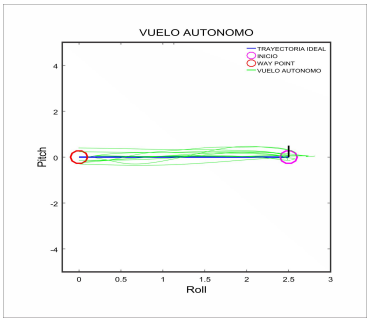
(a) Área de pruebas en Interiores

(b) Área de pruebas en Exteriores

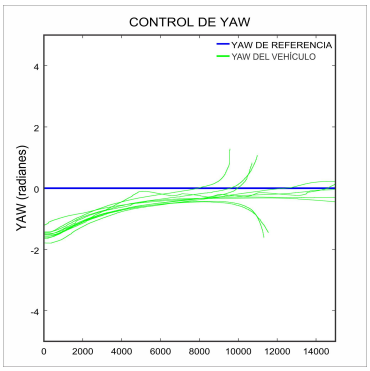


(c) Área de coordenadas generada por el sistema Vicon

Fig. 3. Muestra del área donde se realizaron los experimentos y el área de referencia con el sistema de coordenadas que genera el Vicon



(a) Experimentos de vuelo autónomo en una trayectoria en línea recta



(b) Yaw del vehículo aéreo durante el recorrido hacia el punto (0,0)

Fig. 4. Muestra de los experimentos realizados dentro del laboratorio de robótica donde se puede apreciar la trayectoria recorrida por el vehículo aéreo (imagen izquierda), y la corrección del Yaw desde el momento en que se despega el vehículo aéreo (imagen derecha). Nota las variaciones en las gráficas pueden ser arbitrarias debido al efecto suelo, esto es porque los experimentos se realizaron en un lugar cerrado

En la Fig. 4b se muestra el yaw del vehículo aéreo a lo largo de la trayectoria hacia el punto (0,0), donde se puede notar que el control PID intenta corregir el ángulo del vehículo procurando que éste apunte directamente al punto final. Sin embargo, en algunos experimentos a pesar de que el vehículo apuntaba al punto final, al momento del aterrizaje el ángulo final se veía afectado por la inercia sin afectar la posición final.

4.2. Pruebas en exteriores

El propósito de estas pruebas fue el observar el comportamiento del vehículo aéreo siendo pilotado por nuestro control PID en un ambiente dinámico. Para esta prueba se retiró la carcasa protectora del vehículo aéreo, para así disminuir el peso y el arrastre generado por el viento, con el fin de mejorar la estabilidad y maniobrabilidad. Al igual que en las pruebas en interiores se estableció un área de trabajo, donde se colocaron las cámaras Vicon al rededor del área de trabajo. Posteriormente se calibraron las cámaras. El vehículo se posicionó para todos los experimentos en un punto cercano al (0,0), variando su orientación inicial y se realizaron 3 series de experimentos, donde los puntos finales a alcanzar se ubicaron a 4m, 5m, 6m respecto al punto inicial. Es importante mencionar que en estas pruebas existieron perturbaciones del viento. Sin embargo, nuestro control PID consiguió compensar esas perturbaciones logrando los resultados que se mencionan a continuación.

En la Fig. 5a se muestra una vista superior de la trayectoria seguida por el vehículo aéreo, donde el punto final estaba ubicado a 4m del punto inicial. El promedio del punto de aterrizaje de los 3 experimentos realizados se encuentra a 1.65cm del punto final, cuya desviación estándar es de 0.53cm.

Para la trayectoria mostrada en la Fig. 5b el punto final estaba ubicado a 5m del punto inicial. Los resultados oscilan entre 4.7m y 5.4m, debido a las ráfagas de viento. No obstante el promedio de aterrizaje se encuentra a 5.85cm del punto final, con una desviación estándar de 4.45cm.

La Fig. 5c muestra una vista superior de la trayectoria seguida por el vehículo. Se realizaron 2 experimentos ubicando el punto final a 6 m del punto inicial. Para esta serie de experimentos se obtuvo un promedio de aterrizaje de 5.89cm del punto final cuya desviación estándar fue de 3.57cm.

La Fig. 6a se muestra el yaw del vehículo aéreo a lo largo de las trayectorias hacia los puntos finales. Se puede apreciar que el control PID cumple al corregir el ángulo del vehículo, ya que éste apuntaba hacia el punto final.

La Fig. 6b muestra el yaw del vehículo aéreo recibiendo ráfagas de viento a lo largo de la trayectoria hacia el punto final. A pesar de que en periodos de tiempo cortos la orientación se ve seriamente afectada y que existen pérdidas de información, se puede apreciar que el control intenta mantener al vehículo apuntando hacia la ubicación final.

5. Conclusiones

En este artículo se presentó la combinación de un controlador PID y el sistema Vicon para micro-vehículos aéreos. Esta combinación controla el cabeceo y la traslación de un micro-vehículo aéreo, con el objetivo de que éste pueda navegar entre dos puntos en un espacio de manera autónoma. Para esto, se utilizó el sistema de captura de movimiento Vicon como retroalimentación, el cual opera en ambientes interiores y exteriores. Los resultados obtenidos indican que la metodología aquí presentada es funcional y permite que el vehículo ejecute el vuelo autónomo de manera estable. Los resultados obtenidos fueron satisfactorios a pesar de que el vehículo aéreo era poco estable. Sin embargo, es necesario mejorar el control PID para utilizarlo en condiciones ambientales adversas.

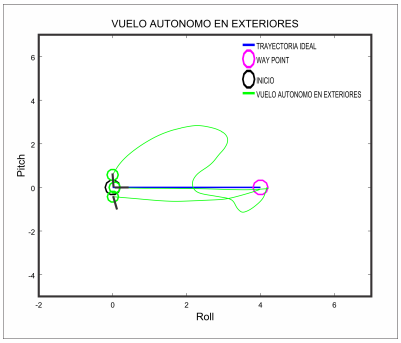
Como trabajo futuro se contempla la implementación de este tipo de control para vehículos aéreos con un sistema de localización eficaz para lograr vuelos autónomos. También cabe recalcar que para que un vehículo aéreo pueda volar de manera autónoma y segura, es necesario implementar un sistema de visión con sensado para que el vehículo aéreo pueda evitar posibles obstáculos.

Agradecimientos Este trabajo fue financiado por la Royal Society-Newton Advanced Fellowship con referencia NA140454.

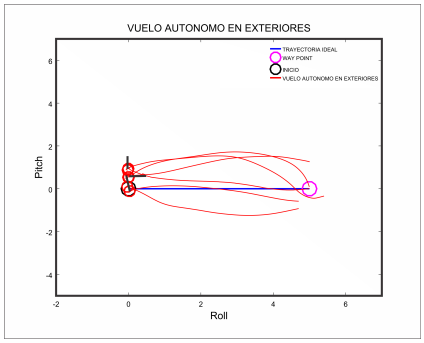
Referencias

1. Vicon Tracker 3 Fast. Flexible. Precise, `http://www.vicon.com/products/software/tracker`
2. Ávila, D., Lorusso, S.F., Pereira, G., Mazza, N., Irribarren, J., Meda, Rodolfo., Ierache, J.: Robótica Situada Aplicada al Control de Vuelo Autónomo de un Cuadricóptero. Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica (ISIER), pp. 3–8, Morón, Buenos Aires, Argentina (2009)
3. Cortés, U., Castañeda, A., Benítez, A., Díaz, A.: Control de Movimiento de un Robot Móvil Tipo Diferencial Robot uBop-32b. En: Congreso Nacional de Control Automático, AMCA 2015, pp. 3–6, Cuernavaca, Morelos, México (2015)
4. Jorge, O.S, Victor, S.: Controlador PID lineal para robots manipuladores considerando saturaciones propias de las etapas de control y de los actuadores. En: Congreso Anual 2009 de la Asociación de México de Control Automático, pp.1-4. Zacatecas, México.
5. E., Gaviria, C.A.: Control PID Multivariable y Modos Deslizantes de un Robot SCARA. pp. 3–5, Popayán, Cauca, Colombia (2009)
6. Garijo, D., López, J.I., Pérez, I.: Control de un vehículo aéreo no tripulado. Madrid, España, pp.116–118 y pp.143–146, S.S.I.I. (2009)
7. Paul, E.I., Pounds, D. R., Bersak, A. M.: Dollar Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control. Autonomous Robots, Volume 33, Issue 1, pp 129–142, August (2012)
8. Núñez, B. D.: Diseño e implementación de un cuadracóptero con sistema de control automático de estabilidad y comunicación inalámbrica de datos utilizando plataformas de hardware y software libre. Costa Rica, pp. 52–64 (2012)

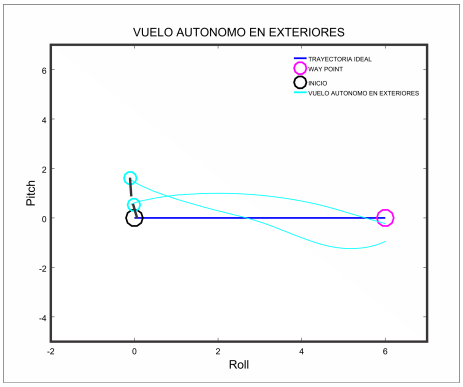
9. González, L. M.: Diseño del sistema de control de un UAV de ala fija para vuelo autónomo en exteriores. Madrid, España, pp. 3–4, Junio (2015)
10. Parada, P, E.: Quadcopter: Construcción, control de vuelo y navegación GPS. Madrid, España, Universidad Carlos III de Madrid, pp. 49–58, Octubre (2012)
11. Melero, C. D: Modelado dinámico y diseño de estrategia de control mediante estimadores para el vuelo autónomo de un quadrotor. Almeria, España, pp. 27–30 y pp. 59–77, Septiembre (2012)
12. Pico, V. A.: Diseño e implementación de un sistema de control para un cuadricóptero. México, Distrito Federal, pp. 17–29 y pp. 39–45, Diciembre (2012)



(a) Experimentos con una trayectoria de cuatro metros

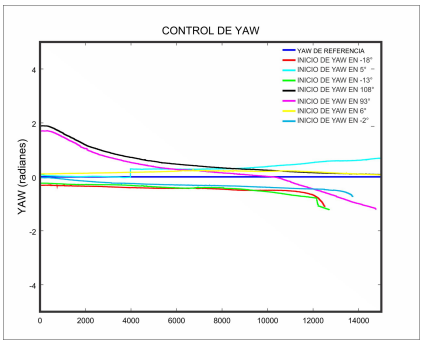


(b) Experimentos con una trayectoria de cinco metros

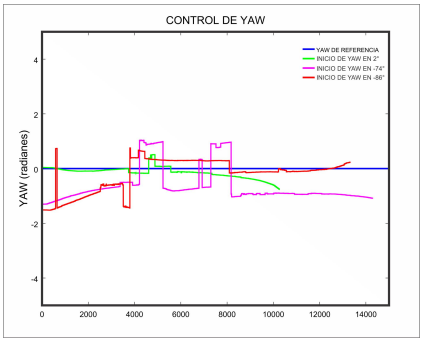


(c) Experimentos con una trayectoria de seis metros

Fig. 5. Muestra de los experimentos realizados en el exterior donde se puede apreciar la trayectoria a distancias diferentes donde es posible ver el cumplimiento del PID



(a) Experimentos del Yaw cumpliendo el ángulo de orientación



(b) Experimentos del Yaw con ráfagas de viento

Fig. 6. Muestra de los experimentos realizados donde se puede apreciar el yaw con y sin las perturbaciones que presentaba el vehículo al entrar en vuelo

DBSCAN modificado con Octrees para agrupar nubes de puntos en tiempo real

Octavio Navarro-Hinojosa, Moisés Alencastre-Miranda

Tecnologico de Monterrey, Campus Santa Fe, Ciudad de México, México

A00967953@itesm.mx, malencastre@itesm.mx

Resumen. Con el auge de sensores de profundidad comerciales, como el Kinect, se crean nuevas oportunidades para desarrollar aplicaciones y sistemas interactivos que utilicen el cuerpo humano. Sin embargo, esos sensores generan una gran cantidad de datos en 3D (nubes de puntos) que tienen que ser procesados, buscando obtener obtener información relevante para aplicaciones específicas. Los algoritmos de agrupamiento, usualmente usados para minería de datos, son útiles para descubrir esa información. Uno de los más conocidos es DBSCAN, que permite generar un número desconocido de grupos en un conjunto de datos, al mismo tiempo que filtra ruido. Sin embargo, también puede ser lento debido al tipo de dato que se use, así como a la búsqueda de datos con características similares. En este trabajo, se propone el uso de DBSCAN para procesar nubes de puntos. Adicionalmente, se propone una modificación del algoritmo, utilizando octrees para acelerar la búsqueda de vecinos que realiza, así como un esquema de particionamiento para que no se tengan que hacer búsquedas de vecinos de todos los puntos de una nube. Con el método propuesto, se logró acelerar considerablemente el procesamiento de nubes de puntos en comparación con el algoritmo original, logrando procesamiento en tiempo real, obteniendo los mismos resultados.

Palabras clave: DBSCAN, particionamiento de datos, estructuras de datos espaciales, datos de profundidad, aplicaciones interactivas.

Modification of the DBSCAN Algorithm with Octrees in order to Cluster Point Clouds in Real Time

Abstract. With the advent of commercial depth sensors, such as the Kinect, new opportunities to develop applications and interactive systems that use the human body are created. However, those sensors capture a large amount of 3D data, known as point clouds, that have to be processed, trying to obtain as much relevant information as possible. Clustering algorithms, normally used in data mining, are useful to

discover that information. One of the most known is DBSCAN, which allows the creation of an unknown number of clusters within a data set, while filtering out noise. Unfortunately, the algorithm can also be slow due to the data type of the data set, as well as because of the search of data with similar characteristics. In this work, the use of the DBSCAN algorithm to process point clouds is proposed. Additionally, a modification of the algorithm is presented: using octrees to decrease the neighbor searching times, and using a data partitioning scheme in order to reduce the number of elements of each search. With the proposed method, the point cloud processing time was considerably accelerated when compared to the original algorithm, achieving real time processing, while obtaining the same results.

Keywords: DBSCAN, data partitioning, spatial data structures, depth data, interactive applications.

1. Introducción

En los últimos años, la introducción de sensores de profundidad ha creado nuevas oportunidades para generar experiencias y aplicaciones interactivas en las que se utiliza el cuerpo humano. El Kinect de Microsoft [14] fue uno de los primeros sensores de profundidad comerciales de acceso general disponibles en el mercado. Aunque su principal uso fue para jugar videojuegos interactivos utilizando el cuerpo [16], éste le permitió a desarrolladores e investigadores explorar el uso de reconocimiento de movimiento en áreas como salud, educación, entretenimiento, arte, robótica, reconocimiento de gestos, reconstrucción 3D, entre otras [12].

Usando un sensor infrarrojo, el Kinect genera un flujo de datos de profundidad que sirve para construir nubes de puntos en tres dimensiones que representan un lugar físico. Dependiendo de la aplicación final, el volumen de datos en una nube que se obtiene de un solo punto en el tiempo puede ser considerablemente alto, hasta un total de 307200 puntos por cuadro [14]. Al desarrollar aplicaciones interactivas, es necesario encontrar métodos que permitan procesar ese volumen de datos en tiempo real, es decir, procesar al menos 30 cuadros por segundo, y que permitan obtener información relevante. Una de las técnicas que logran esto, es el análisis de grupos (clustering analysis).

El análisis de grupos es una técnica en minería de datos que divide un conjunto de datos en grupos, cada uno con características similares, y ayuda a obtener información adicional de dichos grupos. Un algoritmo de agrupamiento descubre esos grupos en los datos al maximizar las similitudes entre un grupo de objetos, y minimizarlo entre los grupos que se generen. Algunos de los campos de aplicación para el análisis de grupos son análisis estadístico, reconocimiento de patrones, procesamiento de imágenes, entre otros [18].

Los algoritmos de agrupamiento se pueden dividir en cuatro tipos: particionamiento, jerárquicos, basados en densidad, y basados en mallas [8]. DBSCAN (Density Based Spatial Clustering of Applications with Noise) es un algoritmo

de agrupamiento basado en densidad. La idea principal detrás del algoritmo es que por cada dato en un grupo, el vecindario dentro de un radio determinado (*eps*) tiene que contener al menos un número mínimo de puntos (*minpts*), es decir, la densidad de un vecindario tiene que exceder un umbral específico [5].

Uno de los problemas de DBSCAN es que debido a su complejidad computacional de $O(n^2)$ [6] no es eficiente para procesar un gran número de datos, como las nubes de puntos que arroja el Kinect, en tiempo real. Otro punto importante es que la mayoría del tiempo que utiliza el algoritmo para el proceso de agrupamiento es usado en las operaciones de búsqueda de vecinos [18].

En el presente trabajo, se tiene como objetivo realizar modificaciones al algoritmo DBSCAN para lograr que procese nubes de puntos, obtenidas de un sensor de profundidad, en tiempo real, y poder utilizarlo en aplicaciones interactivas como videojuegos. Se propone el uso de Octrees (estructuras de datos en las que cada nodo tiene exactamente ocho hijos, que sirven para subdividir un espacio en tres dimensiones) [13] para acelerar la búsqueda de vecinos del algoritmo. De igual forma, se propone un nuevo esquema de particionamiento con el que se reduce considerablemente el espacio de búsqueda del algoritmo, al mismo tiempo que se obtienen los mismos grupos que con el algoritmo original.

El resto de este trabajo se organiza como sigue. La siguiente sección muestra trabajo relacionado. Después se presenta un resumen del algoritmo original, y se analizan sus limitaciones al trabajar con un gran volumen de datos. De igual forma, se presentará la idea básica de los Octrees, y el por qué de su uso para mejorar el desempeño de DBSCAN. Posteriormente, se introduce el algoritmo modificado indicando cómo se integra el uso de octrees, así como el esquema de particionamiento. Después, se mostrarán resultados experimentales para demostrar la efectividad del algoritmo propuesto al aplicarlo para agrupar nubes de profundidad. Finalmente se concluye y se comenta el trabajo a futuro.

2. Trabajo Relacionado

El algoritmo DBSCAN, propuesto por Ester et. al. [5], está basado en la detección de grupos al comparar las densidades entre los puntos de un conjunto determinado. Aunque es conceptualmente sencillo, ha probado ser efectivo para identificar grupos en grandes bases de datos. En los últimos años, muchos trabajos basados en DBSCAN han sido propuestos y aplicados en diferentes áreas. De igual forma, han habido muchos trabajos que buscan mejorar el desempeño del algoritmo.

Zhou et. al. [18] propone varios enfoques para mejorar el desempeño de DBSCAN y poder aplicarlo a bases de datos grandes. Entre ellos, una modificación basada en muestras, otra basada en particionamiento, y un algoritmo paralelo. Arlia et. al. [1] define un algoritmo en paralelo basado en un esquema de esclavo y maestro, donde el maestro se encarga de hacer la asignación de grupos, mientras el esclavo realiza consultas de vecindad sobre un árbol R*. Götz et.al. [8] presentan una versión en paralelo y escalable de DBSCAN. Su trabajo se basa en un esquema de dividir y conquistar, usando técnicas de algoritmos

de agrupamiento basados en celdas. Específicamente, utilizan una hiper-malla para minimizar la búsqueda de vecinos, y para particionar el agrupamiento en subtareas. Finalmente, proponen un esquema de unión para combinar los grupos que se obtuvieron en las subtareas. Thapa et. al. [17] proponen un enfoque con tarjetas de video programables (GPU, por sus siglas en inglés), para procesar DBSCAN en paralelo. Su trabajo permite una mejor escalabilidad de memoria, lo que les permite usar el algoritmo con bases de datos muy grandes. De igual forma, se mejora considerablemente el tiempo de ejecución del algoritmo.

Zhou et. al. [19] introduce un nuevo esquema para agrupar basado en árboles de búsqueda digital, entre ellos quadtrees y octrees. Su enfoque ayudó a reducir los espacios de búsqueda que los algoritmos de agrupamiento usualmente tienen que recorrer para agrupar los datos. Principalmente ayudan a evitar dimensiones elevadas de búsqueda, y eliminar unidades vacías en las que se hubiera tenido que hacer búsquedas. Joshi et. al. [10] proponen un algoritmo basado en DBSCAN, para agrupar polígonos en un espacio. Para agrupar polígonos, se incorporan sus propiedades topológicas y espaciales en el proceso de agrupamiento al usar una función de distancia adecuada para el espacio de los polígonos. Su objetivo es crear grupos compactos de polígonos.

En el trabajo de Borah et. al. [3], se presenta una mejora para DBSCAN basada en un muestreo de los datos de una base de datos. Su método reduce el uso de memoria y el tiempo de ejecución, pero mantiene la calidad de los grupos generados. En [11], Liu propone una modificación a DBSCAN en la que se busca mejorar el tiempo de ejecución de las búsquedas por región del algoritmo, al seleccionar ciertos puntos ordenados sin marcar que están fuera del vecindario de los puntos núcleo, y con ellos expandir los grupos. Este enfoque mejora tanto los tiempos de ejecución como la calidad de los grupos encontrados.

Bianchi y Martinelly [2] utilizan DBSCAN para estimar, en tiempo real, la forma y posición de objetos en una nube de puntos de profundidad. Ya que DBSCAN da buenos resultados, pero es caro computacionalmente y no permite tiempo real, se utilizó un subconjunto de los datos, seleccionado con filtros de imágenes basados en convolución, para acelerar el agrupamiento de los puntos. En el presentado por Doan et. al. [4] se busca hacer una segmentación de objetos encontrados en nubes de puntos de profundidad utilizando DBSCAN para extraer los grupos que representen un objeto. Su objetivo es desarrollar aplicaciones de realidad aumentada, usando sensores de profundidad, para ofrecer una mayor inmersión a los usuarios al modificar los objetos de la escena. Ghosh y Lohani [7] describen un trabajo en el que se analizan nubes de puntos de profundidad obtenidas con tecnología LIDAR aérea, para generar conjuntos de datos densos y precisos de terreno. Se utiliza el método de DBSCAN para separar los datos en grupos distintos de terreno, basándose en reglas topográficas, que serán procesados posteriormente.

Aunque la mayoría de estos trabajos logran reducir considerablemente el tiempo de procesamiento de DBSCAN, solo Götz et.al. [8] logran mejoras en tiempo suficientes para procesar nubes de puntos en tiempo real. Sin embargo, su solución requiere de una gran cantidad de recursos computacionales: sus experi-

mentos y tiempos reportados fueron realizados con 768 núcleos de procesamiento, de un sistema con 2472 núcleos.

El presente trabajo busca mejorar el desempeño de DBSCAN al utilizar octrees para reducir el tiempo de las búsquedas de vecinos, así como mediante el uso de un esquema de particionamiento, para reducir la cantidad de puntos que van a ser procesados. El objetivo es reducir el tiempo de procesamiento de DBSCAN lo suficiente como para poder utilizarlo en aplicaciones en tiempo real.

3. El algoritmo DBSCAN

DBSCAN es un algoritmo de agrupamiento basado en densidad publicado por Ester et al. [5]. Está diseñado para descubrir grupos de forma arbitraria mientras que es capaz de manejar con eficacia el ruido y los valores atípicos. La idea principal es la de encontrar áreas densas y expandirlas recursivamente para encontrar grupos. Un área densa está formada por un punto que tiene al menos *minPts* puntos vecinos, dentro de un radio de búsqueda ε. Esta área densa también es llamada el *núcleo* de un grupo. Se aplica el concepto previo a cada uno de los vecinos, y el grupo se expande. Todos los puntos que no formen un núcleo y que no son "absorbidos" por este proceso de expansión se consideran como ruido. Las siguientes definiciones describen el algoritmo con respecto a sus parámetros ε y *minPts* [8]. En la Figura 1 se pueden ver ilustradas estas definiciones.

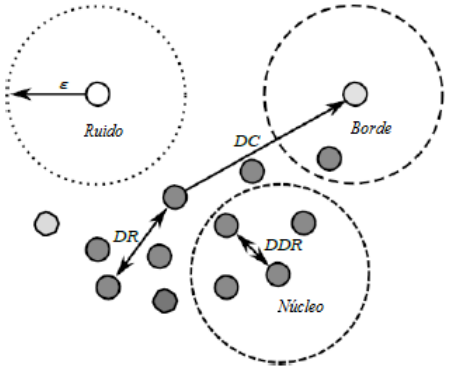


Fig. 1: Agrupamiento con DBSCAN con $minPts = 4$. Reproducido de [8]

Definición 1. Vecindario Epsilon N_ε: El vecindario epsilon N_ε de p denota todos los puntos q de un conjunto de datos X, que tienen una distancia $dist(p, q)$ que es mejor o igual a ε. Es decir $N_\varepsilon(p) = \{q | dist(p, q) \leq \varepsilon\}$. En la práctica, se usa normalmente la distancia euclidiana para evaluar $dist$.

Definición 2. El punto núcleo: p es considerado un punto núcleo si el vecindario epsilon de p contiene al menos $minPts$ puntos incluyéndose a sí mismo: $Core(p) = |N_\varepsilon(p)| \geq minPts$.

Definición 3. Directamente alcanzable por densidad (Directly Density Reachable, DDR): Un punto q es es directamente alcanzable por densidad desde un punto p, si p está dentro del vecindario epsilon de q y p es un punto núcleo: $DDR(p, q) = q \in N_\varepsilon(p) \wedge Core(p)$.

Definición 4. Alcanzable por densidad (Density Reachable, DR): Un par de puntos $p_0 = p$ y $p_n = q$ se llaman alcanzables por densidad si existe una cadena de puntos directamente alcanzables por densidad -$\{p_i|0 \leq i \wedge i < n \wedge DDR(p_i, p_{i+1})\}$-que los une uno al otro.

Definición 5. Punto borde: Los puntos borde son puntos dentro del grupo que usualmente se encuentran en los extremos. Estos no cumplen el criterio de punto núcleo pero aún así se incluyen en el grupo debido a que son directamente alcanzables por densidad. Formalmente, esto se expresa como: $Border(p) = |N_\varepsilon(p)| < minPts \wedge \exists q : DDR(q, p)$.

Definición 6. Conectado por densidad: Dos puntos p y q se dicen conectados por densidad, si hay un tercer punto r, tal que r pueda alcanzar por densidad a p y q: $DC(p, q) = \exists r \in X : DR(r, p) \wedge DR(r, q)$.

Definición 7. Grupo: Un grupo es un subconjunto de un conjunto de datos, donde cada uno de los puntos es alcanzable por densidad a todos los otros del grupo, y que contiene al menos un punto núcleo. Formalmente esto se define como: $\emptyset \subset C \subseteq X$ con $\forall p, q \in C : DC(p, q)$ y $\exists p \in C : Core(p)$.

Definición 8. Ruido: Los puntos que se consideran como ruido, son puntos que no pertenecen a ningún vecindario epsilon, tal que $Noise(p) = \neg\exists q : DDR(q, p)$.

Para encontrar un grupo, DBSCAN empieza con un punto arbitrario p en X y recupera todos los puntos que sean alcanzables por densidad de p, con respecto a ε y *minPts*. Si p es un punto núcleo, se crea un grupo. Si p es un punto borde, no se alcanzaron puntos por densidad desde p, y p se asigna temporalmente como ruido. Posteriormente, el algoritmo procesa el siguiente punto en el conjunto de datos, y el grupo se va expandiendo al ir recuperando puntos alcanzables por densidad al realizar consultas sucesivas de región.

Considerando que DBSCAN no realiza ningún tipo de pre-agrupamiento, y que opera directamente sobre el conjunto de datos, al trabajar sobre conjuntos de datos grandes, el costo de tiempo para hacer las consultas por región e ir haciendo al agrupamiento de todos los puntos es bastante alto. Esto se puede confirmar con el trabajo de Gunawan [9], donde se expone que el algoritmo tiene una complejidad de $O(n^2)$, principalmente debido a la búsqueda de vecinos. Si se usa una estructura de indexado, como son los árboles R*, el desempeño se mejora considerablemente, ejecutándose en $O(n \log n)$.

4. Octrees

Un octree [13] es una estructura de datos en la que cada nodo tiene exactamente ocho hijos. Se usan principalmente para particionar un espacio en tres

dimensiones al subdividirlo recursivamente en ocho octantes, hasta que se cumpla una condición. Para el caso de un conjunto de datos en tres dimensiones, una condición podría ser que se siga subdividiendo hasta que solamente exista un punto por cada nodo; otra condición sería que se siga subdividiendo hasta que los nodos tengan una longitud de lado determinada. Estos son los análogos en tres dimensiones de los quadtrees. Un ejemplo de la construcción de un octree se puede ver en la Figura 2.

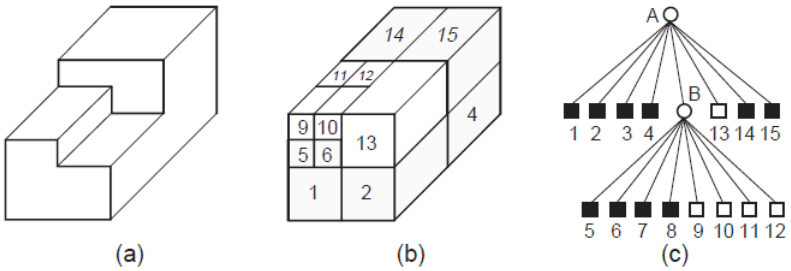


Fig. 2: (a) Ejemplo de un objeto en tres dimensiones; (b) su descomposición en un octree; (c) su representación en forma de árbol. Reproducido de [15]

Un octree es particularmente útil para realizar operaciones sobre conjuntos de datos grandes; una de las principales es la búsqueda de datos basándose en la localización. Esas búsquedas son sencillas de resolver con un octree ya que implican descender por el árbol hasta que se encuentre el objeto que se deseé. Si se desea un vecino de un punto, la búsqueda se continúa en el vecindario del nodo que contiene el objeto [15]. Esta última característica es de principal importancia cuando se considera que el conjunto de datos son nubes de puntos obtenidas de un sensor de profundidad, ya que se están procesando datos en tres dimensiones que representan una posición en el espacio. De igual forma, esta propiedad permite resolver el problema de búsqueda de vecinos que presenta DBSCAN para datos en tres dimensiones.

5. DBSCAN modificado

Considerando que uno de los principales cuellos de botella para el algoritmo es la búsqueda de vecinos, la modificación que se propone busca reducir tanto el tiempo de ejecución que le toma a ese paso, como las veces que se tiene que realizar. Para lograr el primer objetivo, se propone el uso de octrees debido a su eficiencia al indexar y buscar datos en tres dimensiones. Conseguir el segundo objetivo implica hacer una modificación a la forma en la que el algoritmo procesa los puntos. La idea general es la siguiente: Se determinan puntos núcleo con sus respectivos puntos alcanzables por densidad, pero, a diferencia del algoritmo original, no se expanden los grupos que se encuentran. Una vez que se encontraron

todos los grupos sin expandir, se calcula el centroide de estos, y se aplica el proceso de expansión a los centroides, añadiendo todos los puntos de los grupos que son alcanzables por densidad a un nuevo grupo. Esta modificación se explica más a detalle en las siguientes secciones.

5.1. Modificación con octrees

Se hizo una primera modificación a DBSCAN para reducir los tiempos de procesamiento al reemplazar el proceso de búsqueda de vecinos con distancia euclidiana por la búsqueda de vecinos con un octree. Para esto, antes de aplicar DBSCAN, es necesario generar un octree con la nube de puntos inicial. Una ventaja de los octrees es que se puede determinar que se termine la subdivisión del espacio cuando los nodos tengan una cierta longitud. Para el procesamiento de nubes de puntos, se eligió esa condición ya que se pueden tener varios puntos en un mismo nodo sin que posteriores búsquedas de vecinos se vean afectadas, y se logra que el octree resultante no tenga muchos nodos, reduciendo considerablemente el espacio de búsqueda. Otra ventaja de los octrees es que su tiempo de generación y de búsqueda se mantiene en $O(n \log n)$ [15]. Los resultados experimentales demuestran que el uso de octree reduce el tiempo de procesamiento del algoritmo.

5.2. Descripción de los dos pasos de agrupamiento

El algoritmo modificado requiere de dos pasos: búsqueda de puntos núcleo y generación de grupos iniciales, y de expansión de grupos basado en sus centroides. El primer paso es similar al algoritmo DBSCAN original, sin la parte de expansión de los grupos. Para este paso, el radio de búsqueda y los puntos mínimos son los mismos que para el algoritmo original. Es importante notar que para este proceso, sí se tienen que recorrer todos los puntos de la nube, sin embargo, se evita hacer búsquedas de región por cada uno de ellos. El resultado de este paso, que se puede ver en la Figura 3, es una lista que contiene los grupos de los puntos núcleo y los puntos que son alcanzables por densidad de estos. Hay que considerar que se va a tener una gran cantidad de grupos, pero en comparación con los puntos iniciales, el número es mucho menor: esto va a depender de el problema que se esté resolviendo, ya que el radio de búsqueda y los puntos mínimos afectan la cantidad de grupos que se obtienen. El pseudocódigo de este paso se puede ver en el Algoritmo 1.

El siguiente paso consiste en agrupar los grupos que se obtuvieron en el primer paso. Para esto, se obtienen los centroides de los grupos y se construye un segundo octree que será utilizado para hacer las siguientes búsquedas por región. El paso de expansión del algoritmo original se aplica a los centroides obtenidos, con el detalle de que las búsquedas van a ir agregando los puntos de cada uno de los grupos en lugar de los puntos de los centroides. Como se van a ir haciendo búsquedas por región sobre los centroides, el radio de búsqueda tiene que ser dos veces el radio de búsqueda original, para que a partir de cada uno hayan centroides que sean alcanzables por densidad. El resultado final, que

```
Octree = Crear_octree(minPts, nube_puntos)
Def DBSCAN_1(nube_puntos, Octree, eps, minPts):
grupos = lista()
for p en nube_puntos do
    if !visitado(p) then
        grupo = lista()
        marcar_visitado(p)
        grupo.agregar(p)
        vecinos = Consulta_vecinos_octree(Octree, p, eps)
        if vecinos < minPts then
            ruido.anadir(p)
        else
            for q en vecinos do
                if !visitado(q) then
                    marcar_visitado(q)
                    grupo.agregar(q)
                end
            end
            if tamano(grupo) > minPts then
                grupos.agregar(grupo)
            end
        end
    end
end
```

Algoritmo 1: Paso 1: Generación de grupos intermedios.

se puede ver en la Figura 4, son los grupos finales. En el Algoritmo 2 se puede ver el pseudocódigo de este paso.

6. Resultados experimentales

En esta sección se describe la metodología y resultados de los experimentos que fueron realizados para evaluar el algoritmo propuesto. El desarrollo de los algoritmos mencionados fue realizado en un equipo con el sistema operativo Windows 7, que cuenta con un procesador Intel Core i7-3612QM a 2.10GHz, en el ambiente de desarrollo Visual C++ 2013. Se hicieron pruebas sobre el siguiente conjunto de datos: un conjunto de 3600 nubes de puntos de profundidad que se grabó a 30 cuadros por segundo. Las nubes de puntos tienen entre 10000 y 46000 puntos de tres dimensiones, dependiendo de lo que se vea en cada cuadro de la escena. La escena que se grabó consiste de un fondo donde se observan manos y caras haciendo movimientos. Los datos se obtuvieron con el Kinect para Windows y el SDK 1.8.

Debido a que el objetivo es aplicar el algoritmo a una nube de puntos que se obtuvo con el Kinect, el radio de búsqueda que se usa como parámetro del algoritmo se estableció en 40 milímetros. Se eligió ese valor después de hacer un análisis del promedio de distancia entre cada punto y sus 4 vecinos

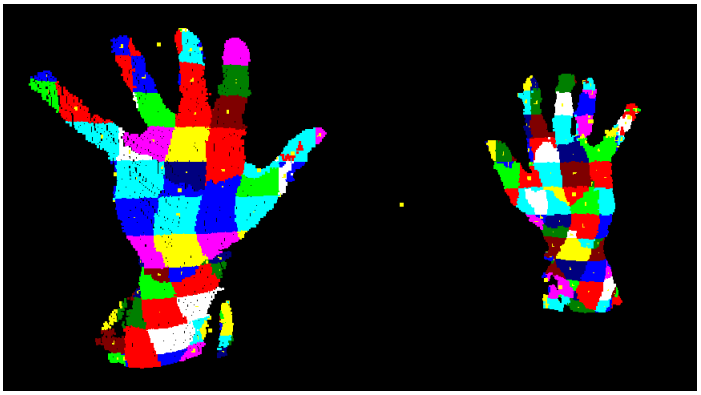


Fig. 3: Resultado de aplicar DBSCAN con un umbral de densidad menor a una nube de puntos. Los colores repetidos no indican que los puntos pertenecen al mismo grupo

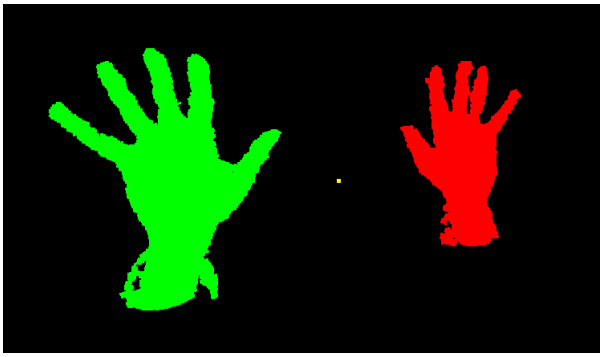


Fig. 4: Resultado de la expansión de los grupos obtenidos en el paso 1

más próximos [18]. El número mínimo de puntos para la primera parte del algoritmo se estableció en 100, mientras que para la segunda parte del algoritmo se estableció en 1000.

6.1. Desempeño del algoritmo

La Figura 5 muestra los promedios de tiempo de aplicar 100 veces los diferentes algoritmos a nubes de puntos con diferente número de puntos en cada una. En cada una de las pruebas se dieron los mismos resultados: El número de grupos fue el mismo, y se agruparon los mismos puntos, así como que se eliminaron los mismos puntos de ruido.

En primer lugar, se puede notar que utilizando el algoritmo original con octrees para realizar las búsquedas de vecinos, se obtiene una mejora considerable en tiempo en comparación con el algoritmo original. Esto se puede ver aún más claramente en la Figura 6, donde se puede ver el aumento de velocidad, en promedio de 970, que se tuvo al aplicar el algoritmo.

```
Centroides = lista()
for grupo en grupos do
    Centroides.agregar(Calcular_centroide(grupo))
end
Octree_2 = Crear_octree(minPts, Centroides)
Def DBSCAN_2(Centroides, grupos_iniciales, Octree_2, eps, minPts):
grupos_finales = lista()
for p en Centroides do
    if !visitado(p) then
        grupo = lista()
        marcar_visitado(p)
        grupo.agregar(Puntos(grupos_iniciales(p)))
        vecinos = Consulta_vecinos_octree(Octree_2, p, 2 * eps)
        for q en vecinos do
            if !visitado(q) then
                marcar_visitado(q)
                grupo.agregar(Puntos(grupos_iniciales(q)))
            end
        end
        if tamano(grupo) > minPts then
            grupos_finales.agregar(grupo)
        end
    end
end
```

Algoritmo 2: Paso 2: Expansión de grupos intermedios para obtener grupos finales.

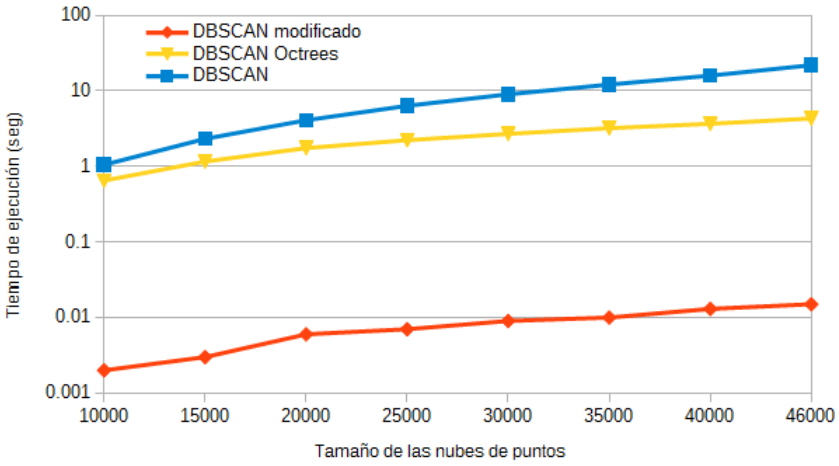


Fig. 5: Comparación de desempeño entre DBSCAN, DBSCAN solo con octrees para realizar las búsquedas, y el DBSCAN modificado

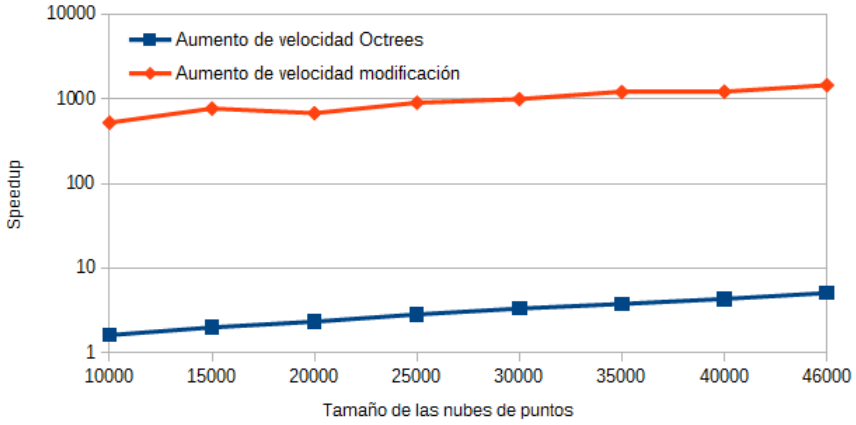


Fig. 6: Aumento de velocidad de los algoritmos

Al aplicar el método propuesto, se obtiene una mejora en tiempo que permite que se pueda usar en una aplicación interactiva, aún con nubes de puntos de gran tamaño. Con los tiempos que se obtuvieron para procesar una nube de puntos de 46000 puntos, se puede lograr un desempeño en tiempo de procesamiento de alrededor de 66 cuadros por segundo. Esto permite que se hagan otros procesos, y aún se pueda alcanzar el procesamiento en tiempo real, de alrededor de 30 cuadros por segundo.

7. Conclusión y trabajo a futuro

En este trabajo se presentó un algoritmo basado en DBSCAN para procesar nubes de puntos en tiempo real. Para lograr el procesamiento de las nubes en tiempo real, se tuvieron que utilizar tanto estructuras de datos espaciales, como son los octrees, como un esquema de particionamiento para reducir el espacio de búsqueda de vecinos. Se modificó el algoritmo original al separar en dos partes el procesamiento de los puntos. La primera parte genera grupos en base a puntos núcleo, y la segunda parte une esos grupos al aplicar la parte de expansión del algoritmo original. Todas las búsquedas de vecinos se realizan con los métodos de búsqueda por región que ofrecen los octrees. Al aplicar el algoritmo propuesto, y comparar los resultados de agrupamiento con el algoritmo original, se pudo observar que se obtenían los mismos grupos.

Como trabajo futuro se planea aplicar el algoritmo en un mayor número de conjuntos de datos, y no necesariamente datos de nubes de puntos obtenidas con algún sensor de profundidad. De igual manera, hay muchos algoritmos que han aprovechado arquitecturas en paralelo, tanto para CPU como para GPU, para acelerar el algoritmo DBSCAN. Se planea explorar dichos algoritmos para integrarlos con el algoritmo propuesto y mejorar los tiempos de procesamien-

to obtenidos. Específicamente, se plantea el uso de GPU para poder hacer el procesamiento de nubes de puntos de mayor tamaño al utilizado en éste trabajo.

Referencias

1. Arlia, D., Coppola, M.: Experiments in parallel clustering with dbscan. In: Euro-Par 2001 Parallel Processing, pp. 326–331. Springer (2001)
2. Bianchi, L., Martinelli, A.: A clustering approach to object estimation, featuring image filtering prototyping for dbscan in virtual sets. In: Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on. pp. 751–756. IEEE (2007)
3. Borah, B., Bhattacharyya, D.: An improved sampling-based dbscan for large spatial databases. In: Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on. pp. 92–96. IEEE (2004)
4. Doan, N.H., Pham, D., Dinh, T.B., Dinh, T.B.: Restoring surfaces after removing objects in indoor 3d point clouds. In: Proceedings of the Fourth Symposium on Information and Communication Technology. pp. 189–197. ACM (2013)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovering in Databases and Data Mining. vol. 96, pp. 226–231 (1996)
6. Gan, J., Tao, Y.: Dbscan revisited: Mis-claim, un-fixability, and approximation. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. pp. 519–530. ACM (2015)
7. Ghosh, S., Lohani, B.: Development and comparison of aerial photograph aided visualization pipelines for lidar datasets. International Journal of Digital Earth 8(8), 656–677 (2015)
8. Götz, M., Bodenstein, C., Riedel, M.: Hpdbscan: highly parallel dbscan. In: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments. p. 2. ACM (2015)
9. Gunawan, A., de Berg, M.: A faster algorithm for DBSCAN. Ph.D. thesis, Master (2013)
10. Joshi, D., Samal, A.K., Soh, L.K.: Density-based clustering of polygons. In: Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on. pp. 171–178. IEEE (2009)
11. Liu, B.: A fast density-based clustering algorithm for large databases. In: Machine Learning and Cybernetics, 2006 International Conference on. pp. 996–1000. IEEE (2006)
12. Lun, R., Zhao, W.: A survey of applications and human motion recognition with microsoft kinect. International Journal of Pattern Recognition and Artificial Intelligence 29(05), 1555008 (2015)
13. Meagher, D.J.: Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer. Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory (1980)
14. Microsoft: Kinect for windows programming guide. https://msdn.microsoft.com/en-us/library/hh855348.aspx, accesado: 2015-04-11
15. Samet, H.: Spatial Data Structures: Quadtree, Octrees and Other Hierarchical Methods. Addison Wesley Reading (1989)

16. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM 56(1), 116–124 (2013)
17. Thapa, R.J., Trefftz, C., Wolffe, G.: Memory-efficient implementation of a graphics processor-based cluster detection algorithm for large spatial databases. In: Electro/Information Technology (EIT), 2010 IEEE International Conference on. pp. 1–5. IEEE (2010)
18. Zhou, A., Zhou, S., Cao, J., Fan, Y., Hu, Y.: Approaches for scaling dbscan algorithm to large spatial databases. Journal of computer science and technology 15(6), 509–526 (2000)
19. Zhou, X.H., Wang, H.B., Zhou, D.R., Meng, B.: Data clustering algorithm based on digital search tree. In: Machine Learning and Cybernetics, 2003 International Conference on. vol. 3, pp. 1757–1761. IEEE (2003)

Sistema inmune artificial para estegoanálisis de imágenes JPEG

José de Jesús Serrano-Pérez, Moisés Salinas-Rosales, Nareli Cruz-Cortés

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Ciberseguridad, Ciudad de México,
México

Resumen. La esteganografía es la técnica de ocultar la información digital quizás más utilizada en la actualidad. Existen numerosos reportes sobre su uso exitoso para ocultar código malicioso dentro de objetos multimedia que infiltran software dañino (malware) en diversos dispositivos electrónicos, evitando su detección por los controles correspondientes. Una vez que el malware ha alcanzado su destino, otro software extrae el código incrustado y ejecuta un ataque. Este tipo de malware es llamado *estegomalware*. El estegoanálisis es la contramedida de la esteganografía, que se refiere al estudio de las técnicas que permite la detección de *esteganogramas*. En el estegoanálisis moderno se han utilizado diversas técnicas de la Inteligencia Artificial. En este trabajo exploramos el uso de un paradigma con inspiración biológica llamado Sistema Inmune Artificial (SIA) para detectar imágenes en formato JPGE alteradas con esteganografía. Además, proponemos el uso de la ondeleta de Haar para la definición del vector de características que mejor describa a la imagen bajo análisis. Los experimentos ejecutados arrojaron resultados prometedores que prueban que la detección realizada por nuestra propuesta son comparables, y ocasiones mejores, que otros trabajos representativos del estado del arte.

Palabras clave: Esteganografía, estegoanálisis, sistemas inmunes artificiales, clasificación, reconocimiento de patrones.

Steganalysis Based on an Artificial Immune System for JPEG Images

Abstract. Steganography is one of the most used hiding information techniques today.Recently, the use of steganography techniques has been reported very successful to hide malicious code inside, apparently innocuous, multimedia objects, in order to infiltrate malware into organizations and personal devices, avoiding malware detection controls. Once the embedded malware has reached its destination, another software extracts the embedded code and performs the attack. This new kind of malware is called *stegomalware*. Steganalysis is the countermeasure to steganography, and it is a set of techniques that allows the detection of

these *steganograms*. In modern steganalysis different Artificial Intelligence techniques have been employed, but very few have proposed solutions based on Bio-inspired and Evolutionary Computing. In this work we present a steganography detection method based on an Artificial Immune System (AIS) to detect JPEG images altered with steganography. We propose the use of Haar Wavelets in order to extract a characteristic feature vector that best describe the analyzed image. The experiments performed shown promising results that could prove that a classification system made with AIS could perform same and in some cases better.

Keywords: Steganography, steganalysis, artificial immune systems, wavelets, classification, pattern recognition.

1. Introducción

La esteganografía es una rama de las técnicas de ocultamiento de la información, cuyo uso permite establecer un canal de comunicación encubierto y seguro, donde solo las entidades involucradas son conscientes de él, para eso se selecciona un elemento inocuo y común que pasará fácilmente desapercibido, ademas de que incluso a una inspección superficial no se le encontrara rareza alguna. Si bien el uso de estas tecnicas remota a las primeras civilizaciones. En la actualidad su uso sigue vigente y tiene una presencia muy fuerte y dominante en el mundo digital, donde es utilizado para burlar sistemas de censura en países totalitarios para poder comunicarse con el mundo exterior, así como para la exfiltración de información sensible, etc. En la figura 1 podemos ver de que se compone un sistema esteganografico, el emisor genera una llave que codificara el mensaje en un objeto por medio de una función embeber, el resultado es un objeto encubierto o esteganograma, al llegar al receptor, este utiliza la misma llave con la que se codifico el mensaje con una función extracción. Ejemplo de una herramienta esteganográfica, es Outguess [7]

Ejemplo de esto se ve en la figura 1, donde el mensaje a enviar se codifica con una llave y el objeto que servia como encubrimiento, cuando viaje por un canal de comunicación, se le conocerá como esteganograma, una vez que llegue a su destino el receptor, con la misma llave con la que se codifico el mensaje, se extrae y obtiene la información oculta en ese esteganograma.

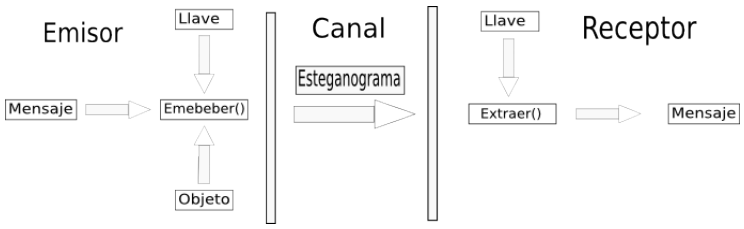


Fig. 1. Modelo un sistema esteganográfico

Sin embargo se le ha encontrado un nuevo uso a la esteganografía, donde aprovechando las propiedades de ocultamiento, se ha comenzado a utilizar para ocultar código malicioso o virus informáticos, donde una vez que el estganograma ha llegado a su destino final, son extraídos del objeto donde se ocultaban vulnerando así los controles antimalware y de control de contenido. A este nueva forma de implementar *malware* se le conoce como *stegomalware*; ejemplos de esto han sido Lurk [10], Stegosploit [8] y ataques de filtración al sistema operativo Android [11].

El estegoanálisis es la contraparte de la esteganografía y se encarga de la detección de información oculta en un un grupo de objetos dados, para eso emplea diversas técnicas de disciplinas computaciones como análisis de imágenes y señales, inteligencia artificial y reconocimiento de patrones, entre otras. Se han propuesto varias técnicas para poder detectar este tipo de objetos.

Utilizando distintas técnicas de las diferentes disciplinas mencionadas. En este trabajo se propone un sistema inmune artificial (SIA) para la detección de esteganogramas, la información se utiliza para entrenar el sistema se extrae por medio del análisis de ondeletas de Harr en imágenes.

Las principales aportaciones en este trabajo son las siguientes:

1. La adaptación de un SIA para detectar esteganogramas.
2. El uso de ondeletas de Harr para caracterizar las imagenes y se pueda identificar aquellas que son esteganogramas.

Una de las principales ventajas del uso de esta ondeleta es que son de cálculo rápido y permiten obtener una representación compacta de las imágenes a caracterizar. Los resultados obtenidos en los experimentos muestran que la propuesta es factible y competitiva.

En la sección 2 se describe lo que es el estegoanalisis, los diferentes tipos y métodos existentes, en la sección 3 se menciona lo que es el sistema inmune artificial y como se desarrolla una aplicación utilizando este paradigma computacional, en la sección 4 se describe la propuesta del SIA como clasificador de esteganogramas, en la sección 5 se detalla la metodología que se sigue para desarrollar el sistema y los resultados obtenidos, posteriormente en la sección 6 se discuten los resultados obtenidos y se menciona el posible trabajo a futuro para realizar

2. Estegoanálisis

No hay un método único y formal para crear un *esteganograma* por lo que el estegoanálisis requiere de múltiples técnicas de diferentes áreas para poder reconocer aquellos objetos que pudieran ser esteganogramas [1]. Esto hace el problema de detección interesante y complejo por las implicaciones practicas que conlleva, partiendo de la experimentación, hallar el método para poder encontrar la o las características que permiten la detección de los objetos. Seguido de esto es necesario escoger un método de clasificación que en base a la información obtenida de la imagen pueda clasificar de manera correcta estos objetos y determine si contienen información oculta o no.

2.1. Técnicas de estegoanálisis

Dentro del estegoanálisis se definen 2 tipos de detectores, los dedicados y los universales, el primero es un tipo de detector especifico para un tipo de técnica en particular, su ventaja es el de un porcentaje detección mayor comparado con los detectores universales, pero con la desventaja de que debe de saber *a priori* que tipo de técnica fue utilizada en el objeto a analizar. Por otra parte, los detectores universales son aquellos que detectan mas de 2 tipos de técnicas esteganográficas, su desventaja es que tienden a tener una tasa de detección menor que la de los detectores dedicados, pero no requieren saber que técnica esteganográficas fue empleada en el objeto a analizar.

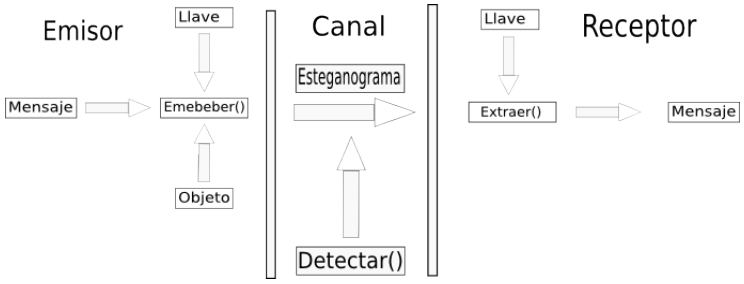


Fig. 2. Detector con enfoque de guardia pasivo

A su vez cualquiera de los dos tipos se divide en dos enfoques, activo y pasivo, el primer enfoque consiste en modificar todos los elementos que viajan en el canal de comunicación, de manera que no afecte su uso legitimo (ejemplo la visualización de una imagen) pero cuando se requiera recuperar la información, esto no sea posible. Aunque supone una solución final al problema de la detección de la esteganografía su implementación es meramente teórica. Para el enfoque pasivo representado en la figura 2 este se encarga de analizar cada objeto por medio de una función de detección, tal que dada una probabilidad se determina si, un objeto es un esteganograma o no. Este trabajo se enfoca en los detectores de tipo dedicados con enfoque pasivo.

2.2. Estado del arte

El trabajo realizado por Dumitrescu,2003 [4] realiza un trabajo de detección con enfoque dedicado, detectando en los bits menos significativos variaciones anormales en elementos multimedia como audio y vídeo, los clasifica usando analisis estadistico obteniendo resultados de detección entre el 87 % y 99 %. Un trabajo mas reciente realizado por Chen, 2009 detectando la herramienta esteganográfica JPEG-Steg [3] donde usa modelos mixtos lineares como método de extracción de características en imágenes JPEG y como método de clasificación utiliza redes neuronales, obteniendo asi resultados de detección del 98 % al 100 %,

el trabajo de Sindhu, 2008 [9] extrae las características de imágenes BMP por medio de análisis estadísticos en los bits menos significativos y por medio de análisis estadísticos de primer orden detecta las variaciones en el conjunto de características, obteniendo resultados de clasificación entre el 92 % y 99 %

3. Sistemas inmunes artificial

El sistema inmune artificial (SIA) son un nuevo paradigma computacional donde se busca emular el sistema inmune humano, de modo que se pueda adaptar como un método de clasificación, estos sistemas tienen la propiedad de adaptabilidad y auto-mantenimiento. A pesar de ser un reciente paradigma computación comparado con otros métodos de clasificación como maquinas de soporte vectorial, aprendizaje maquina o redes neuronales, ha demostrado obtener resultados similares a los métodos con mayor tiempo de desarrollo e investigación. De manera formal un SIA se define de la siguiente manera: Ün sistema adaptativo, inspirado por la teoría de inmunología y la observación de funciones inmunes, principios y modelos, que son aplicados a la solución de problemas"[12].

Sus usos principales son para la solución de problemas de clasificación y optimización, las áreas donde mas son utilizados por mencionar algunas:

- Seguridad computacional,
- Optimización de funciones numéricas,
- Aprendizaje maquina,
- Bio-informatica,
- Detección de virus,
- Minería de datos.

3.1. Desarrollando una arquitectura de SIA

Hasta la fecha no se ha desarrollado una metodología general para los SIA, pero Castro y Timmis [2] sugieren una arquitectura para su desarrollo, donde se definen los siguientes puntos:

- Primero definir la representación para los componentes del sistema,
- Un conjunto de mecanismos para evaluar la interacción entre los individuos,
- Un conjunto de procedimientos de adaptación que gobiernen las dinámicas del sistema.

Para la representación de los componentes del sistema se han empleado representaciones como cadenas binarias y vectores de valores reales, para nuestro caso emplearemos los vectores de valores reales que obtuvimos del análisis por medio de las ondeletas de Harr. Para poder realizar la interacción, un método de evaluación debe ser definido para poder determinar la afinidad de los elementos que componen el sistema, ya que estaremos usando vectores de valores reales, vamos a usar distancia euclidiana para dicho fin:

$$A = \sqrt{\sum_{i=1}^{L} (Ab_i - Ag_i)^2}. \quad (1)$$

Para definir el conjunto de procedimientos que gobernaran el sistema usaremos el algoritmo de seleccion negativa, que menciona los siguientes pasos:

1. Definir el *self* y el no *non-self*,
2. Crear anticuerpos que sean diferentes del *self*,
3. Entrenar y probar los anticuerpos con un conjunto de patógenos de entrenamiento,
4. Una vez que se tiene un numero definido de anticuerpos maduros, probar el sistema contra un conjunto de pruebas.

Para que se pueda cumplir el algoritmo se tiene que cumplir la siguiente definición formal sobre los conjuntos *self* y *non-self*: Dado el espacio $\sum^L$ y el conjunto que define *self* $S \subset \sum^L$ definimos el conjunto *non-self* $N \subset \sum^L$ ser el complemento $N = \sum^L \backslash S$ tal que $\sum^L = S \cup N$ y $S \cap N = \emptyset$.

De manera practica puede ser que dicha definición no pueda cumplirse, así que se tiene que buscar que la representación de los elementos del sistema corresponda a la definición previamente descrita

4. Propuesta

El trabajo propuesto es desarrollar un sistema de detección de esteganogramas usando un clasificador que utilice un SIA en imágenes JPEG alterados con la herramienta Outguess, utilizando un sistema inmune artificial usando un algoritmo de clasificación negativa, así como la identificación de los rasgos que mejor describen a las imágenes alteradas con Outguess, para eso usaremos los coeficientes obtenidos del análisis por ondeletas de Harr. El desarrollo se describe de la siguiente manera:

1. Crear dos repositorios con las mismas imágenes, uno alterarlo con una herramienta esteganográfica y el otro dejarlo intacto, estos serán nuestros repositorios de entrenamiento.
2. Realizar la extracción de características por medio del análisis de ondeletas de Harr para cada uno de los repositorios creados.
3. Utilizando el algoritmo de selección negativa del sistema inmune artificial , desarrollar el clasificador y entrenarlo con los datos de los repositorios de entrenamiento.
4. Probar el clasificador contra un repositorio de imagenes de prueba.

A continuacion de describira cada uno de los puntos mencionados en el desarrollo empezando por la extracción de características:

4.1. Extracción de características

Para poder desarrollar un detector de esteganogramas JPEG se requiere primero obtener un vector de rasgos que nos permita caracterizar bien el esteganograma que queremos identificar, ya que nuestro interés está en las imágenes

JPEG, escogeremos una técnica que nos permita obtener la información más relevante de la imagen, para ello existen varias formas de obtener esta información, dentro de las más utilizadas está el uso de análisis por ondeletas, cuya ventaja es su rápido cálculo y un conjunto de coeficientes que describen la imagen de manera precisa, existen dentro de la familia de ondeletas varios tipos de ellas, como por ejemplo Daubechies y Harr. Las ondeletas de Harr en particular se caracterizan por su rapidez de cálculo y uso para describir imágenes [6]. Para esto se usa la segunda transformada de ondeletas de Harr la cual se describe de la siguiente manera [5]:

$$\overline{f} = \begin{pmatrix} f_{0,0} & f_{0,\frac{1}{2}} \\ f_{\frac{1}{2},0} & f_{\frac{1}{2},\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} s_{0,0} & s_{0,1} \\ s_{1,0} & s_{1,1} \end{pmatrix}. \tag{2}$$

Considerar la función $\overline{f}$ que es la aproximación de una función f y s_0, s_1 los valores de la señal a analizar (siendo la amplitud y longitud de la señal respectivamente).

$$\begin{pmatrix} s_{0,0} & s_{0,1} \\ s_{1,0} & s_{1,1} \end{pmatrix} \rightrightarrows \begin{pmatrix} \frac{s_{0,0}+s_{0,1}}{2} & \frac{s_{0,0}-s_{0,1}}{2} \\ \frac{s_{1,0}+s_{1,1}}{2} & \frac{s_{1,0}-s_{1,1}}{2} \end{pmatrix}. \tag{3}$$

Después de que se hayan aplicado un número de iteraciones exitosas (usualmente n cuando el tamaño de la señal es 2^n) obtenemos la siguiente matriz

$$\begin{pmatrix} CA & CH \\ CV & CD \end{pmatrix}, \tag{4}$$

donde cada coeficiente describe lo siguiente:

- **Cambios Ponderados** (CA) : Incluye las propiedades globales de la señal/imagen analizada.
- **Cambios Horizontales** (CH) : Incluye la información sobre las lineas horizontales ocultas en la señal/imagen.
- **Cambios Verticales** (CV): Incluye la información sobre las lineas verticales ocultas en la señal/imagen.
- **Cambios Diagonales** (CD): Incluye la información sobre las lineas diagonales ocultas en la señal/imagen.

Cuando se realiza el analisis en una imagen a color se obtienen 48 valores en total, 12 corresponden a los coeficientes en CA, 12 en CH, 12 en CV y 12 en CD, esto por que es una imagen a color, nosotros solo utilizaremos los 3 últimos coeficientes que son CH, CD y CD, siendo un total de 36 coeficientes que forman nuestro vector de caracterización, el cual representamos en la tabla 1.

En la figura 3, se muestra un análisis de dos imágenes iguales en sus cuatro coeficientes, las imágenes del primer renglón corresponden a un esteganograma, las imágenes correspondientes al segundo renglón son la misma imagen pero sin contenido oculto, visualmente hay diferencias notorias cuando comparamos ambas imágenes en el coeficiente horizontal, vertical, diagonal y el ponderado. De esta manera se ejemplifica que es posible crear un vector de características significativamente descriptivo usando los coeficientes obtenidos por la ondeleta de Harr.

Tabla 1. Vector de caracterización

CH			CV			CD		
C_0	...	C_{11}	C_{12}	...	C_{23}	C_{24}	...	C_{36}

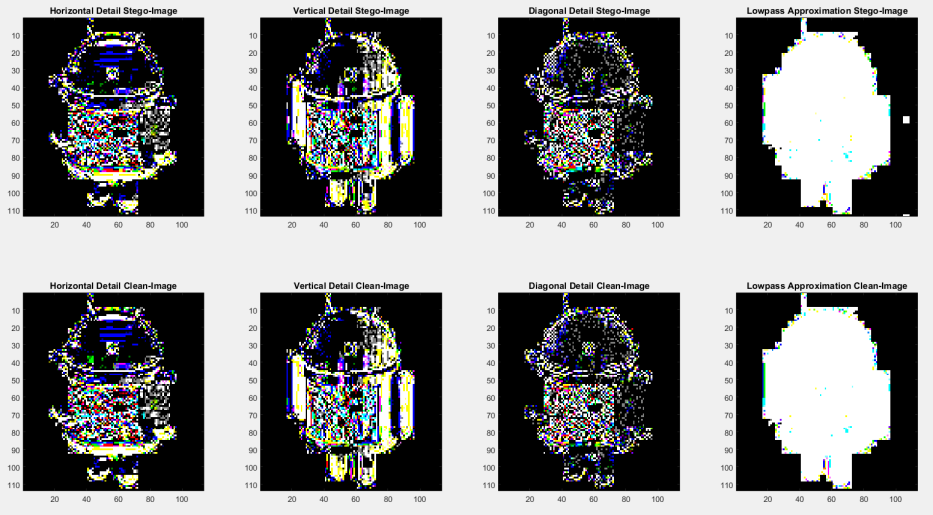


Fig. 3. Análisis visual de una imagen limpia y la misma con contenido oculto usando ondeletas de Harr

4.2. Clasificador SIA para esteganogramas

Siguiendo la arquitectura de desarrollo mostrada en la sección 3.1, se describe lo siguiente:

1. Para la representación de los componentes en el sistema se tienen 3 entidades, el conjunto del *self* que son todas aquellas imágenes que no tienen contenido oculto, el conjunto del *non-self* que son las imágenes con contenido oculto y los detectores que identificaran a los invasores, que serán todos aquellos elementos que no estén definidos en los conjuntos anteriores, cada uno está representado por un vector de características , es decir por un vector formado por 36 valores numéricos.
2. El mecanismo que evalúa la interacción entre estos componentes es la medida de afinidad que se haya seleccionado, en este caso la medida esta dada por la ecuación (1) de distancia euclidiana
3. El procedimiento de gobernará la dinámica del sistema sera el algoritmo de selección negativa también definido en la sección 3.1

4.3. Definiendo el SIA

Procedemos a definir el *self* y *non-self*:

Tabla 2. Composición del anticuerpo

CH			CV			CD			Afinidad con *non-self*	Afinidad con *self*
C_0	...	C_{11}	C_{12}	...	C_{23}	C_{24}	...	C_{36}	$A(Ab_i, NS_i)$	$A(Ab_i, S_i)$

- El *self* serán todos los vectores de características que correspondan al repositorio de imágenes limpias
- El *non-self* serán todos los vectores de características que correspondan al repositorio de imágenes con contenido oculto.
- Una vez definidos los repositorios inicializamos los anticuerpos, que tendrán las mismas características que los vectores antes mencionados, estarán compuestos por 36 valores aleatorios reales en un rango máximo y mínimo definido por los valores del conjunto de entrenamiento, tendrá ademas dos valores adicionales, uno que determine su afinidad con el repositorio de *self* y *non-self*, esta afinidad es la distancia euclidiana entre el anticuerpo y el elemento del repositorio, esto esta representado en la tabla 2.

4.4. Entrenamiento del SIA

Una vez que estén inicializados los repositorios, seguimos a la fase de entrenamiento, donde cada nuevo anticuerpo creado ,al que le llamaremos inmaduro, se le calcula su afinidad con el elemento correspondiente en el conjunto *self* y *non-self*, esto es con la premisa de que para cada elemento en el repositorio del *self* y *non-self* se tiene un anticuerpo. Cada anticuerpo pasa por la siguiente prueba:

- Mientras $numAbM < n$
  - Si $A(Ab_i, NS_i) < A(Ab_i, S_i)$
    - Destruir y crear un nuevo anticuerpo.
  - Si $A(Ab_i, NS_i) > A(Ab_i, S_i)$
    - Agregar el anticuerpo al conjunto de anticuerpos maduros,
    - $numAbM = numAbM + 1$.

$A(Ab_i, NS_i)$ es la afinidad entre el anticuerpo con el elemento correspondiente en *non-self* y $A(Ab_i, S_i)$ es la afinidad con el elemento correspondiente en el *self*. Este proceso se repite hasta haya n anticuerpos maduros($numAbM$), a cada iteración se le denomina ciclo de vida, de manera adicional este entrenamiento evita que los anticuerpos sean auto-reactivos y detecten elementos que no deben identificar.

4.5. Probando el SIA

Una vez que todos los detectores hayan madurado, procedemos a probar el sistema, todos los anticuerpos analizan todos los elementos de prueba y se define la siguiente regla de solución:

- Si $A(Ab_i, In_j) \geq A(Ab_i, NS_i)$ y $A(Ab_i, In_j) > A(Ab_i, S_i)$

- Clasificar como esteganograma.

donde $A(Ab_i, In_j)$ es la afinidad entre un anticuerpo contra un elemento de prueba, el conjunto In es el repositorio de pruebas representado de la misma manera que los conjuntos del $self$ y *non-self.*

5. Experimentos

Para comprobar la funcionalidad del sistema se siguio la metodologia descrita, inciando con la creación del repositorio inicial, donde se utilizaron 1000 imágenes JPEG en RGB, todas las imagenes tienen un tamaño fijo de 512x512 pixeles con una profundidad de color de 24 bits y una tasa de compresión de aproximadamente 26.2313, al tener los 2 repositorios se obtuvieron 2000 imágenes en total, el archivo embebido en el repositorio de imágenes alteradas con Outguess fue un código malicioso en Javascript, los vectores de características fueron almacenados en archivos CSV y gráficados para su primer análisis como se puede ver en la figura 4.

Tabla 3. Resultados obtenidos del SIA en la detección de Outguess

Detección Outguess				
Coeficiente Utilizado	Exhaustividad	Mejor Precisión	Peor Precisión	Precisión en Promedio
CH	100 %	94 %	77 %	86.24 %
CV	100 %	87 %	72 %	80.7 %
CD	100 %	84 %	65 %	75.82 %

Si nosotros utilizáramos los todos los coeficientes como un solo vector de características, difícilmente podríamos diferenciar entre lo que queremos identificar, en nuestro caso al querer implementar el SIA, no se cumpliría la caracterización del *self* y *non-self*, por lo que hemos decidido dividir el vector de características en 3 vectores que contienen los CH, CV y CD por separado, cada uno de estos grupos esta representado en la figura 4, cada vector se compone de 12 coeficientes en total. Para el SIA utilizamos 1000 detectores. Observamos que conforme se va a acercando al numero de detectores deseados, requiere mas ciclos de vida. Una vez que maduraron los 1000 detectores se procedió a probar el sistema, para eso empleamos un nuevo repositorio de 200 imágenes distintas a las utilizadas en la fase de entrenamiento, este repositorio se compone de 100 imágenes limpias y 100 alteradas con Outguess, nuevamente se les extrae los coeficientes para su clasificación en el sistema.

Es importante mencionar que el sistema se probo para los tres casos, cuando usamos los datos de CH, CV y CD. Se ejecuto el programa 50 veces y los resultados obtenidos para cuando se utiliza el sistema usando los 3 coeficientes por separado se muestran en la tabla 5, la linea negra, denotada por los puntos (+), representa los resultados obtenidos de usar CH, la linea roja, denotada por

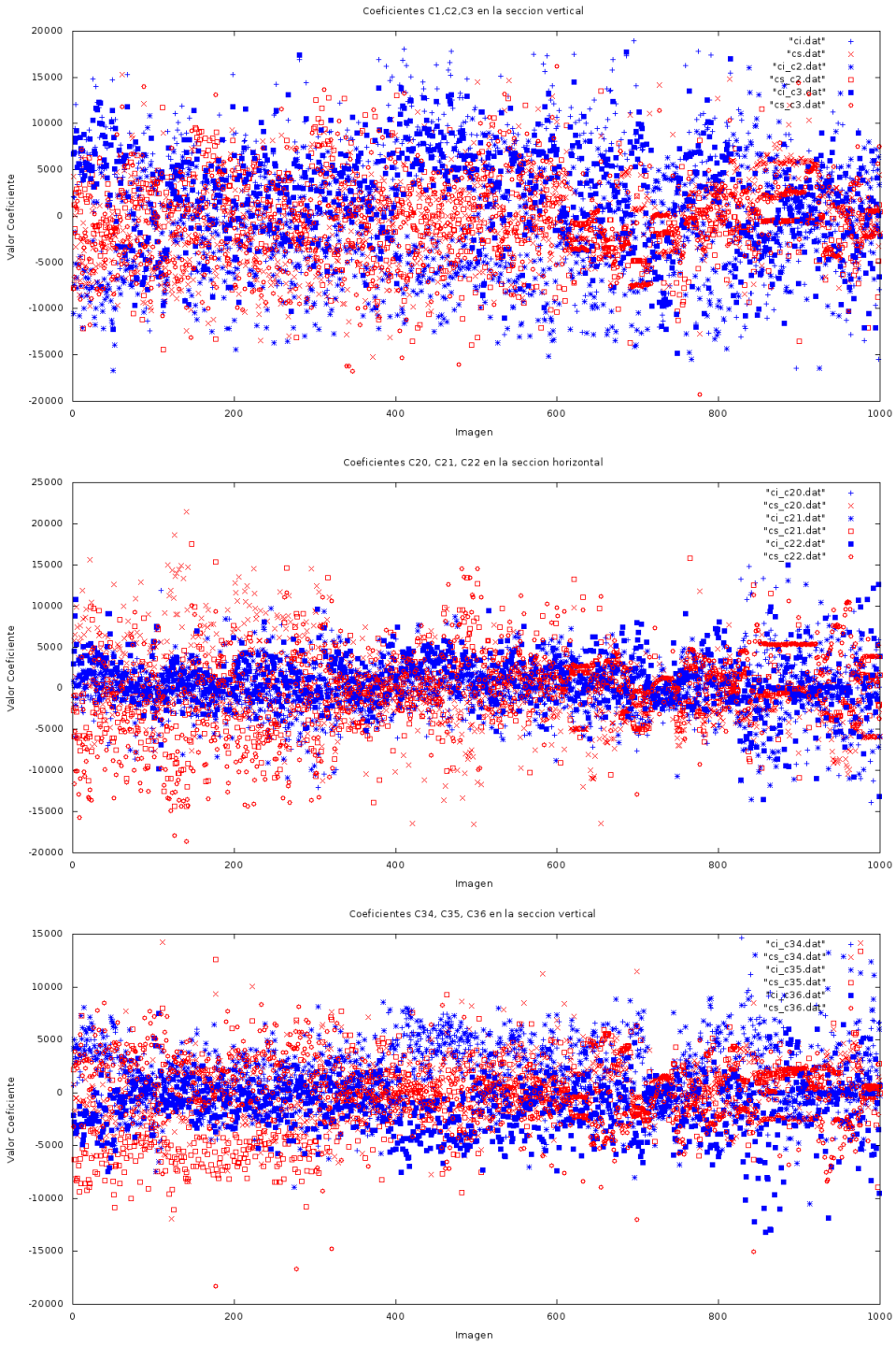


Fig. 4. Representación gráfica de los coeficientes

los puntos(■), de usar CV y la linea azul, denotada por los puntos (•), de usar CD. En la tabla 3 mostramos los resultados obtenidos. Con los coeficientes ya extraidos de las imagenes, el tiempo promedio del sistema en realizar un corrida es de 15 segundos.

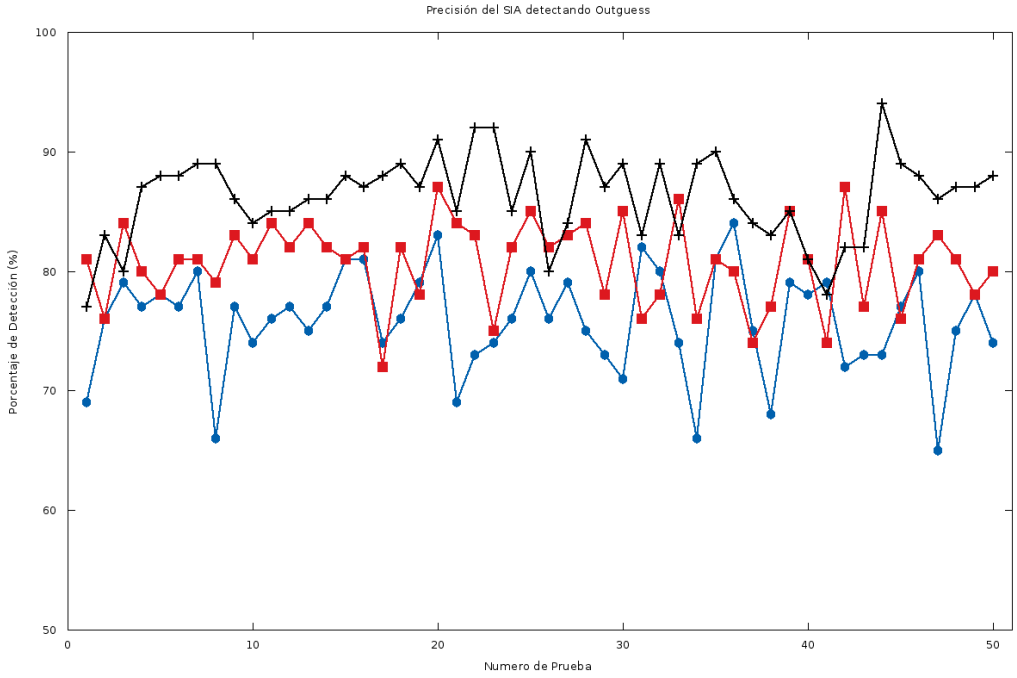


Fig. 5. Evolución de los detectores respecto a cada uno de los coeficientes utilizados

6. Discusión y conclusiones

Con los resultados en la tabla 3 y con los coeficientes gráficados en la figura 5, nos damos cuenta que si influye mucho la selección del vector de características que mejor caracterice a Outguess, vemos que el mejor coeficiente para la caracterización es CH, ya que se obtuvo una precisión del 94 % como mejor y una ponderada del 86.24 %. Comparándolos con los trabajos mencionados en el estado del arte, los resultados son competitivos. De esta manera demostramos que un clasificador de esteganogramas para Outguess basado en un sistema inmune artificial obtiene resultados competitivos iguales o mejores que los trabajos reportados en la literatura y que tiene un gran potencial como herramienta

de estegoanálisis, así como el uso de ondeletas de Harr para la caracterización de estos objetos con contenido oculto. Trabajo a futuro sería el probar este sistema para otras técnicas esteganográficas como F5 y Steghide, así como su implementación con un enfoque de detector universal, así como un análisis de sensibilidad de los parámetros del algoritmo.

Agradecimientos. Se agradece el apoyo del Instituto Politécnico Nacional a través de los proyectos de investigación SIP-20161234 y SIP-20160314.

Referencias

1. Böhme, R.: Advanced Statistical Steganalysis. Information Security and Cryptography, Springer Berlin Heidelberg (2010)
2. Castro, L.: Artificial immune systems : a new computational intelligence approach. Springer, London New York (2002)
3. Chen, M.C., Roy, A., Rodriguez, B.M., Agaian, S.S., Chen, C.L.P.: An application of linear mixed effects model to steganography detection. In: Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on. pp. 1782–1786 (Oct 2009)
4. Dumitrescu, S., Wu, X., Wang, Z.: Detection of lsb steganography via sample pair analysis. IEEE Transactions on Signal Processing 51(7), 1995–2007 (July 2003)
5. Nievergelt, Y.: Wavelets made easy. Birkhauser, New York (2013)
6. Piotr Porwik, A.L.: The haar–wavelet transform in digital image processing:its status and achievements. MG&V 13(3) (2004)
7. Provos, N.: Defending against statistical steganalysis. In: Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10. SSYM'01, USENIX Association, Berkeley, CA, USA (2001), http://dl.acm.org/citation.cfm?id=1251327.1251351
8. Sha, S.: Stegosploit: Hacking With Pictures (May 2015), https://conference.hitb.org/hitbsecconf2015ams/sessions/stegosploit-hacking-with-pictures/
9. Sindhu, S.S.S., Renganathan, R., Raman, P.J., Kamaraj, N.: Stegohunter: Steganalysis of lsb embedded images based on stego-sensitive threshold close color pair signature. In: Computer Vision, Graphics Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on. pp. 281–288 (Dec 2008)
10. Stone-Gross, B.: Malware Analysis of the Lurk Downloader (Agust 2014), http://www.secureworks.com/cyber-threat-intelligence/threats/malware-analysis-of-the-lurk-downloader/
11. ThreatSolutions: Android malware employs steganography? Not quite... (January 2012), https://www.f-secure.com/weblog/archives/00002305.html
12. Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. Theoretical Computer Science 403(1), 11–32 (2008), http://www.sciencedirect.com/science/article/pii/S0304397508001059

Clasificación de estímulos visuales para control de drones

Eduardo Zecua, Irving Caballero, José Martínez-Carranza, Carlos A. Reyes

Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, México
corichiedu@ccc.inaoep.mx, pirving01@ccc.inaoep.mx, carranza@inaoep.mx, kargaxxi@inaoep.mx

Resumen. Los experimentos con grabaciones Electro-encefalográficas (EEG) registran oscilaciones del potencial de membranas neurales. Estas señales representan un porcentaje elevado de la actividad cerebral tanto del pensamiento como del movimiento corporal. Por lo anterio, en este trabajo se explora el uso de EEG para la clasificación del movimiento ocular, detonado por la observación de algún estímulo visual,de tal forma que los movimientos reconocidos puedan ser utilizados para controlar un dron. Para esto se generó una base de datos que se divide en dos grupos: entrenamiento y prueba, generada de 5 sujetos con una media de edad de 28 años. Se procesó la señal tomando los valores estadisticos que representaran los cambios más significativos y generando los vectores característicos de cada una de las muestras. Una vez entrenada la red neuronal con estos parámetros, se procede a clasificar nuevas estancias y, dependiendo de la clasificación, se manda una instrucción o punto de ubicación a un dron controlado mediante un controlador proporcional integral derivativo (PID), el cual utiliza estimaciones de posición del dron obtenidas a través de un sistema de captura de movimiento VICON.

Palabras clave: EEG, FFT, redes neuronales, drones, PID.

Visual Stimuli Classification to Control Drones

Abstract. Experiments with Electro-encephalographic recordings (EEG) capture oscillations of the potential of neural membranes. These signals represent a high percentage of brain activity in both, thought and body movement. Therefore, in this study we explore the use of EEG for classification of eye movement, triggered by the observation of a visual stimuli, such that, the recognized movements can be used to control a drone. To achieve this, a database is generated and divided into two groups: training and testing, generated with five test subjects with an average age of 28 years. The signal was processed by taking the statistical values that represent the most significant changes and generating the characteristic vectors of each of the samples. Once the neural network was trained with these parameters, it proceeds to classify new instances and, depending on the classification, an instruction or location point is sent to a drone

controlled by an Proportional Integral Derivative controller (PID), which estimates the position of the drone through a motion capture Vicon system.

Keywords: EEG, FFT, neural networks, drones, PID.

1. Introducción

Los vehículos aéreos no tripulados (VANTs) o *drones* actualmente están siendo utilizados para distintos propósitos entre los cuales se encuentran la vigilancia, esfuerzos militares para proveer seguridad, reconocimiento táctico, entre otras [12]. Por otra parte, los experimentos con grabaciones electroencefalográficas (EEG) están siendo utilizadas, cada vez más, como índices del proceso de trabajo de la memoria a lo largo de una variedad de tareas que involucran la operación de un VANT [14].

Las señales de EEG recolectadas en el cráneo humano son fluctuaciones de potenciales eléctricos que reflejan actividad en las estructuras cerebrales subyacentes, particularmente en la corteza cerebral debajo de la superficie del cuero cabelludo. Las oscilaciones que produce el EEG se clasifican en función de su relación con la estimulación y pueden ser espontáneas. Los datos que se observan sugieren una relación positiva entre los estímulos inducidos por estados estacionarios de potenciales evocados visualmente y de la actividad cerebral [1].

Cuando un estímulo sensorial corto se presenta, como una luz parpadeante o un movimiento del antebrazo de una persona,se produce una perturbación en las señales del EEG, la cual inicia después de un pequeño retraso del evento inicial (estímulo) y se esparce por alrededor de medio segundo o menos. Los cambios en la amplitud de la señal, debidos a la perturbación, son pequeños (a lo mucho, unos pocos micro-voltios), y no se aprecian a simple vista dentro de las líneas de actividad cerebral.

La gente puede responder únicamente a una pequeña cantidad de información sensorial presente en cualquier momento. Por ello, la selección de la información es necesaria para facilitar los problemas computacionales introducidos por el enorme número de señales presentes en las superficies sensoriales y para asegurarse que la gente responda a un estímulo que sea relevante a los objetivos de la investigación[3]. Aunque muchos estudios han investigado la atención visual, la atención puede ser centrada en otros atributos (o dimensiones). La gente puede atender a, o "mirar hacia", cierta información visual específica (por ejemplo, un sombrero rojo que lleva un amigo entre una multitud). En este sentido, fijar la mirada en un atributo, mejora la precisión en la detección visual o discriminación de tareas [3].

Motivados por lo anterior, en este trabajo se describen resultados sobre el uso de EEG para llevar a cabo el control de un dron. Para esto, los EEG se utilizan para clasificar el movimiento ocular, el cual se genera a partir de la observación de estímulos visuales. Una vez que el movimiento ocular es reconocido, estos se

traducen con comandos de control para ejectura 5 tareas: despegar; viajar hacia un punto A en el espacio; hacia un punto B; o hacia un punto C; y aterrizar. Los resultados obtenidos en este trabajo indican que los movimientos oculares se reconocen con un 71.2 % de precision. Este porcentaje sin duda puede ser mejorado, no obstante, este porcentaje habla de la viabilidad de utilizar la metodología descrita en este trabajo, la cual involucra el uso de EEG para reconocer el movimiento ocular derivado de un estimulo visual y que puede integrarse en un sistema de control para drones.

Clasificar el movimiento ocular a partir de un estimulo visual es de interés en éste trabajo ya que sería deseable identificar ciertos tipos de moviento ocular, y no sólo eso, si no tambien gestos, muecas,o algún otra seña que permita detectar un posible estado de alerta del piloto mientras vuela el dron. Ésto último a raíz de posibles escenarios donde el piloto observa una situacion de peligro, pero no le da tiempo de usar el control remoto para controlar adecuadamente el vehículo. Sin embargo, si la situación de riesgo se considera como un estímulo visual que es observado por el piloto, dicha situación podría ser reconocido con un sistema basado en la metodología que se presenta en este trabajo, y por ende utilizar dicho reconocimiento para enviar un paro de emergencia al dron o algún comando que le permita alejarse del peligro.

De este modo y con el objetivo de describir con detalle el sistema propuesto, éste artículo se ha organizado de la siguiente manera: el trabajo relacionado se presenta en la sección 2; el sistema y sus componentes principales son descritos en la sección 3; los resultados son presentados y discutidos en la seccíon 4; y finalmente las conclusiones se desglozan en la sección 5 y agradecimientos en la sección 6.

2. Trabajos relacionados

Científicos de la Universidad de Texas en San Antonio (UTSA) están tratando de convertir la ciencia ficción en realidad desarrollando la tecnología que permitirá a los soldados controlar remotamente un VANT únicamente con sus mentes. Seis profesores de diferentes de departamentos de la universidad trabajan en distintos proyectos que tienen que ver con el estudio de la interacción cerebro-máquina.

Pero UTSA no es la primera universidad con drones controlados con actividad cerebral. En 2013, el profesor Bin He de la Universidad de Minnesota fue la primera persona en mostrar el vuelo de un pequeño dron cuadrucóptero a través de un aro de globos, completamente controlado con una gorra con 64 sensores de electrodos colocada en la cabeza de una persona. Para volar el dron remótamente, el piloto imaginaba un puño. si se imaginaba un puño con la mano izquierda, el dron se desviaba a la izquierda. Las señales se enviaban de forma inalámbrica desde la computadora hacia el dron, para lo cual, primero decodificaba las señales cerebrales enviadas por la gorra y las retransmitía en forma de comandos que el dron debía seguir.

La dinámica del EEG h sido utilizada para examinar los procesos cerebrales involucrados en tareas como detección visual de un objetivo [10] y rastreo visiomotora [8]. Para el año 2010, se había alcanzado un concenso sobre la mejor aproximación para examinar los datos del EEG[9], pero, en particular, dos anchos de banda de frecuencia han recibido más atención. Se ha encontrado que la actividad en el rango alfa, frecuentemente disminuye con el incremento de la dificultad de la tarea, mientras que lo opuesto se ha observado en la actividad registrada en el rango teta, particularmente en los sitios de los electrodos de la línea media frontal[5].

Esperimentos electrofisiológicos han mostrado que las neuronas en la corteza visual de los humanos sincronizan sus disparos a la frecuencia de luz parpadeante, originando respuestas en el EEG las cuales muestran la misma frecuencia que el estímulo parpadeante [13].

Varios autores han analizado la relación entre la frecuencia EEG y el desempeño de diferentes tareas. En general, a una frecuencia baja del EEG se le ha relacionado a una reacción más larga en el tiempo (Reaction Time) o a un mayor número de errores[15]. La hipótesis más común fue que la relación de desempeño del EEG fue modulada por el nivel de alerta[11]. Está fuera de toda duda que el nivel de alerta produce cambios tanto en el EEG como en el desempeño. Sin embargo, reportes previos de trabajos con niños han mostrado una correlación positiva entre la potencia delta del EEG en reposo, y el tiempo de respuesta (RT) y la relación de error en atención visual y tareas de memorización llevadas a cabo en diferentes sesiones.

3. Descripción del sistema

Para el desarrollo del sistema se emplearon varios equipos descritos en las secciones 3.1 y 3.2. Todos los procesos fueron realizados en una computadora con procesador Intel i7 de cuatro núcleos con 8 Gb de memoria RAM y sistema operativo Ubuntu 14.4 operando con el sistema operativo robótico (ROS). Un segundo equipo con un procesador AMD A10 con 12 Gb de RAM con sistema operativo Windows 10 y operando con Matlab 2015b. Todo el sistema se divide en dos partes: la primera se refiere a la adquisición de los datos del EEG incluyendo su clasificación y, la segunda etapa implica el control del dron utilizando un controlador PID para moverlo a la posición especificada dada la clasificación de las señales.

3.1. Sistema de posicionamiento VICON

El sistema de traqueo Vicon es un sistema de ubicación y seguimiento a partir de cámaras monoculares que, a través de luz infrarroja, envía un haz de luz a marcadores especiales con forma esférica y con un recubrimiento reflejante en el cual rebota la luz en todas direcciones hacia las cámaras. Dependiendo de la cantidad de luz reflejada y la ubicación de cada uno de los marcadores, cada cámara hace una triangulación regresando la ubicación de cada marcador

en el espacio con respecto a una referencia propuesta durante la calibración del equipo y con precisión milimétrica. Estos marcadores se colocaron en el cuerpo del dron de manera que no estorbaran en el vuelo y pudieran ser localizados en todo momento por el sistema de traqueo.

3.2. Dron BEBOP

Para el desarrollo de este proyecto se utilizó el dron BEBOP de la marca PARROT, el cual cuenta con una cámara monocular de tipo ojo de pescado y 4 hélices. Éste es controlado mediante WiFi a través de computadoras o celulares y cuenta con un sensor ultrasónico ubicado en parte inferior que mide la distancia entre él y el suelo lo que permite que se mantenga en vuelo de manera estable. Este modelo cuenta con protecciones para poder operarlo en interiores y exteriores (1) .

Fig. 1. BEBOP operando en vuelo

3.3. Control PID

Un controlador PID (Proportional Integral Derivativo) es un mecanismo de control genérico sobre una retrolimentación de bucle cerrado, ampliamente usado en la industria para el control de sistemas. El PID es un sistema que recibe un error calculado a partir de la salida deseada menos la salida obtenida; y su salida es utilizada como entrada en el sistema que queremos controlar. El controlador intenta minimizar el error ajustando la entrada del sistema.

El controlador PID consiste de tres parámetros distintos: el proporcional, el integral, y el derivativo. El valor Proporcional depende del error actual. El Integral depende de los errores pasados y el Derivativo es una predicción de los errores futuros. La suma de estas tres acciones es usada para ajustar al proceso por medio de un elemento de control como la posición de una válvula de control o la potencia suministrada a un calentador.

Cuando no se tiene conocimiento del proceso, históricamente se ha considerado que el controlador PID es el controlador más adecuado. Ajustando estas tres variables en el controlador PID, puede proveer una acción de control diseñado para los requerimientos del proceso en específico. La respuesta del controlador puede describirse en términos de la respuesta del control ante un error, el grado el cual el controlador sobrepasa el punto de ajuste, y el grado de oscilación del sistema. Nótese que el uso del PID para control no garantiza control óptimo del sistema o la estabilidad del mismo.

Para el sistema de control del Dron se utiliza un PID que calcula el error entre el valor medido y el valor deseado. Esto se aplicó cuando se le asignaba un punto al dron al que tuviera que acercarse, midiendo con el sistema Vicon la posición y calculando el error (distancia entre la posición actual y la deseada). El PID ajusta los valores para acercarse al punto lo más rápido posible con un amortiguamiento en la disminución de la velocidad del vehículo conforme se va acercando al punto deseado. El método fue implementado para el control del desplazamiento del dron y para el giro en su propio eje.

Este modelo se divide en 3 partes:

- La parte proporcional, ajusta la velocidad dependiendo de la distancia entre el punto, esto se refiere a que mientras la distancia entre la posición actual y la deseada sea grande la velocidad del vehículo también será grande y viceversa.
- La parte integral va acumulando el error, esto se refiere a que mientras más tiempo tarde el vehículo en llegar al área deseada, éste acumula los valores para ir incrementado la velocidad en cierta medida.
- La parte derivativa mide la diferencia del error actual y error pasado, mide la proporción de cambio en cada uno de los instantes, lo cual ayuda al sistema a acelerar desde el principio puesto que es cuando el error es mas grande.

El ajuste de las ganancias de estos sistemas no es trivial y requiere ser modificado dependiendo de las condiciones de cada sistema, así como las condiciones de su entorno, ya que el dron es perturbado por el moviendo de sus propias hélices en vuelo.

3.4. Adquisición de señales EEG

Las frecuencias del EEG tradicionalmente se han clasificado en diferentes bandas. La actividad Delta (1.5 - 3.5 Hz) es la principal característica del sueño, pero puede estar presente durante la atención a procesos internos como cálculos mentales y memorización[6]. La actividad Teta (4 - 7 Hz) puede reflejar una función portera del flujo de información a través del hipocampo y los circuitos de estructuras objetivo. La actividad Alfa (8 - 13 Hz) es más que una frecuencia espontánea y es un prototipo de procesos dinámicos que gobiernan un gran conjunto de funciones cerebrales integrativas. Los patrones Alfa puden ser espontáneos, inducidos o evocados por estímulos, movimientos o relaciondos a la memoria. La cuarta actividad es la Gama (25 - 100 Hz) la cual se ha teorizado que podrían estar implicadas en el proceso de percepción consciente [2].

Para la adquisición de las señales se utilizó la diadema de la marca Emotiv modelo EPOC, la cual consta de 14 canales distribuidos en la corteza craneal como se muestra en la figura (2). El sistema tiene una frecuencia de muestro de 128 SPS y se conecta a una computadora de manera inalámbrica a través de una conexión USB. El software utilizado para guardar todas las pruebas realizadas es el Emotiv TestBench de la marca Emotiv.

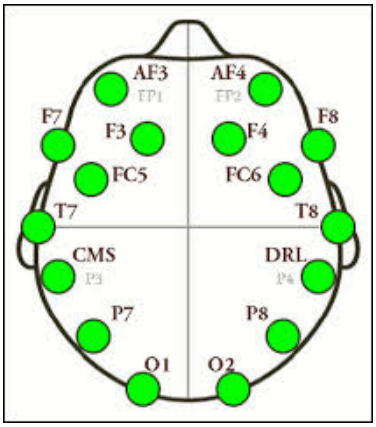


Fig. 2. BEBOP operando en vuelo

La base de datos se creó con cinco sujetos con una media de edad de 28 años. Las muestras se recopilaron mientras ellos estaban sentados y mirando una pantalla con una cruz roja en el centro y fondo blanco para que éstas tuvieran la menor cantidad de ruido al mover los ojos por distraccion de manera involuntaria como se aprecia en la figura (3). A los sujetos se les pidió que realizaran un movimiento ocular determinado (mirar hacia arriba, mirar hacia abajo, mirar a la izquierda, mirar a la derecha, parpadear) y cada vez que llevaban a cabo el movimiento, éste se marcaba con una etiqueta diferente en las lecturas del EEG.

3.5. Procesamiento de la señal

Una vez adquiridas las señales estas se guardaban con la extensión. EDF, para poder leer los datos y poder procesarlos se creó un programa en MATLAB que leyera cada una de las filas y descompusiera este formato en una matriz con los valores de cada uno de los 14 canales por separado.

De la matriz obtenida se seleccionan los canales más característicos para el movimiento ocular, estos se encuentran en la parte frontal arriba de los ojos con la diadema EMOTIV tenemos 4 canales (AF3, AF4, F7, F8) [4] que se encuentran cercanas a esta área. De estos cuatro canales se obtienen los valores estadisticos correspondinetes a los valores maximos y minimos que componen

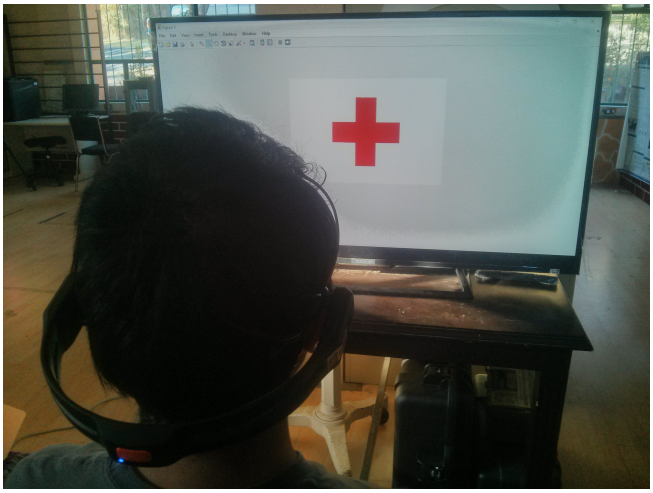


Fig. 3. Montaje de display con cruz roja en el centro

a la señal en el instante en que se generó el movimiento ocular, estos tiene variaciones dependiendo de como se genera el movimiento [7].

3.6. Entrenamiento de la red neuronal

Las pruebas del experimento se dividieron en dos partes, la primera consistía en recopilar muestras para el entrenamiento de una red neuronal creada con ayuda del toolbox de Matlab, siendo una red neuronal de tipo feed-fordward con 5 neuronas en la capa entrada, 8 neuronas en la capa de oculta y 5 neuronas en la capa de salida. Se usó el algoritmo de aprendizaje *trainbr* el cual se utiliza para ajustar los pesos en la fase de entrenamiento, consistiendo en un proceso de regularización bayesiana y fijando en 500 el número de épocas de entrenamiento, con el fin de ajustar los pesos de la red de manera óptima para la clasificación.

Es importante hacer mención que los datos de entrada se etiquetaban con cada uno de los movimientos oculares, proceso que se llevó a cabo mediante un programa generado en Matlab que enviaba los marcadores a través de un clic del ratón para especificar el instante en el que se realizó la acción.

En la segunda parte, se procesaron las muestras para la etapa de evaluación pasándolas por la red neuronal entrenada sin etiquetas, con el objetivo de cuantificar el porcentaje en que las nuevas instancias eran clasificadas correctamente.

3.7. Sistema ROS

Para el control todo el sistema se utilizó el sistema ROS, este es un software especializado en el control de robots el cual funciona con nodos que son lanzados desde códigos en C++ o Python, con la ventaja de poder publicar en cada suscribirse a cada uno de ellos. Este sistema se utilizó para controlar el dron,

el modelo que se utilizó para este experimento fue el BEBOP de la marca PARROT, este se conecta mediante wifi a una computadora o teléfono, para el control se creó un programa que se suscribiera al nodo de clasificación, una vez que este nodo publicara algún mensaje, dependiendo de la clasificación se toma la acción de despegar, aterrizar o volar el dron a una determinada área. La retroalimentación del dron se realizó el área donde este opera con el equipo VICON para saber su posición y la distancia de cada uno de los 3 puntos marcados.

4. Resultados y experimentos

La evaluación del clasificador se llevó a cabo con la base de datos de prueba, este conjunto contenía nuevas instancias que no había visto el clasificador. Una vez pasando por la red neuronal, ésta las clasificaba conforme había aprendido en la fase de enteramiento. Se calculó un arreglo de 5 números formados en columnas como se muestra en el Cuadro 1, cada uno de estos números corresponde al peso de la medida a la instancia que puede pertenecer cada vector. Para hacer la elección de la clase a partir de estos valores, se toma el número mayor y la clase final está dada por la posición de este número, por ejemplo, se puede observar en la columna P1 que el número mayor corresponde a la clase 1. Para construir la tabla se tomaron 10 muestras para poder observar cómo opera el sistema. Estos valores se tomaron al azar de la datos de prueba que cosiste en la combinación de las muestras de los sujetos.

Una vez que se tienen los resultados de pasar todas las muestras de prueba por la red neuronal para ser clasificados, éstas se comparan con los valores de la clase a la que realmente pertenecen. Dichos resultados se presentan en la Matriz de Confusión mostrada en la tabla (2). Cada fila representa la clasificación hecha por la red y las columnas representa la clasificación correcta. En la última fila y columna se muestran los porcentajes de las clasificaciones correctas hechas por el sistema, teniendo el total de aciertos en la celda de la última fila y columna. La posición (1,1) de la tabla nos indica el número de elementos que clasificó correctamente el sistema de la clase 1, en la posición (1,2) nos indica cuantos elementos de la clase 1 fueron clasificados como la clase 2 y así sucesivamente.

Tabla 1. Resultados obtenidos de la clasificación de 10 estimulos. El valor más alto corresponde a la clasificación realizada por el clasificador

Clasificador	P.1	P.2	P.3	P.4	P.5	P.6	P.7	P.8	P.9	P.10
Clase 1	0.8111	0.7856	0.1636	0.1058	0.0008	0.2229	0.5004	0.0359	0.5828	0.2852
Clase 2	-0.5137	0.0866	0.5636	-0.0775	-0.1450	0.1178	0.0919	-0.1162	-0.5676	0.4474
Clase 3	-0.1255	0.1834	-0.0006	0.9726	0.0090	-0.1550	0.1632	0.0138	0.2187	0.1632
Clase 4	0.7191	0.0412	-0.0136	-0.1768	1.1190	-0.0426	0.1715	1.0535	-0.0550	-0.0060
Clase 5	0.1196	-0.0991	0.2896	0.1682	0.0215	0.8565	0.0750	0.0175	0.8018	0.1107

Tabla 2. Matriz de confusión

Clases	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	
Clase 1	29	1	1	2	5	67.9
Clase 2	5	17	0	0	8	56.7
Clase 3	3	0	16	0	4	69.6
Clase 4	1	0	0	28	0	96.6
Clase 5	3	4	3	0	19	65.5
	66.7	77.3	80.0	93.3	52.8	71.2

En los 10 casos de prueba mostrados en el cuadro 1, la prueba 2 (P.2) y la prueba 7 (P.7) fueron erróneas ya que las clases calculadas fueron "1.$^{\text{en}}$ ambas, sin embargo las clases correctas debían ser "2.$^{\text{en}}$ ambos dos casos.

Para el control del vehículo las instrucciones se mandaron mediante un arreglo de 3 números, el primer digito contenía la instrucción de despegar, este se obtuvo de la clasificación 2 (mirar hacia arriba); el segundo digito especificaba a cuál de los 3 puntos pre-establecidos tenía que llegar, obtenido de las clasificaciones 3,4,5 (mirar a la izquierda, mirar a la derecha o parpadear); el ultimo digito contenía la instrucción de aterrizar, obtenido de la clasificación 1 (mirar hacia abajo); estas posibles acciones se describen en el Cuadro (3). Estos se escogieron de esta manera por lo inusual de su comportamiento en un estado normal. Debido a que es más factible parpadear, esta acción se tomó como un valor intermedio para ir a un punto en el espacio. Para que el sistema pudiera empezar a operar es necesario que el sujeto mire hacia arriba para poder despegar el dron. El sistema está diseñado para leer los tres parámetros antes mencionados en este orden. Si recibiera cualquier otro estimulo antes de despegar este no lo tomaría en cuenta. Una vez llevada a cabo la clasificación de los movimientos oculares y poder ejecutar el control del dron, estos valores son impresos en un archivo de texto (.TXT) para, posteriormente, ser leídos por un nodo de ROS y, dependiendo de la clase escrita, el dron realice las acciones predeterminadas. Para las pruebas se realizaron 10 experimentos los cuales se muestran el Cuadro (3).

Tabla 3. Clasificación de 10 experimentos realizados. La primer fila indica el número del experimento. La segunda fila se refiere a la primera acción (despegar). La tercera fila es la acción de movimiento a un punto definido mediante el control PID. La cuarta fila es la acción de aterrizar

1	2	3	4	5	6	7	8	9	10
2—2	2—2	2—2	2—2	2—2	2—2	2—2	2—2	1—2	2—2
4—4	3—3	4—4	1—5	5—5	5—5	4—4	5—5	4—4	4—4
1—1	1—1	1—1	1—1	1—1	4—1	1— 1	1—1	4—1	1—1

El cuadro (3) se divide en columnas las cuales separan cada uno de los experimentos. Cada una de las filas (de arriba hacia abajo) representan los valores ordenados que tomaron en cada uno de los instantes, cada una de estas contiene

dos números, el primero indica el valor que fue capturado en el clasificador y el segundo es el valor de la clasificación real, este último se anexa a la tabla (4). para comprar cada una de las instrucciones.

Tabla 4. Acciones llevadas a cabo de acuerdo a la clasificación de estímulos

Clase	Estímulo	Acción
Clase 1	Mirar abajo	Aterrizar
Clase 2	Mirar arriba	Despegar
Clase 3	Mirar derecha	Mover a punto 1
Clase 4	Mirar izquierda	Mover a punto 2
Clase 5	Parpadeo	Mover a punto 3

5. Conclusiones y trabajo futuro

En este trabajo se han presentado resultados de un sistema basado en el uso de EEG para el reconocimiento de movimiento ocular detonado por un estímulo visual y que, al reconocerse, se utiliza para enviar comandos de control a un dron. Las oscilaciones que produce el EEG pueden ser clasificadas en función de su relación con la estimulación. En este caso, los estímulos correspondieron al movimiento ocular de los sujetos de prueba y dichas oscilaciones se clasificaron en cinco clases distintas con las cuales se pudo accionar un dron con cinco diferentes acciones, dependiendo de la clase enviada. El clasificador obtenido tiene una precisión del 71.2 %, lo cual fue suficiente para los propositos de este trabajo y en medida de que el dron llevo a cabo las acciones determinadas sin mayor dificultad.

No obstante, para el trabajo a futuro se tiene considerado el mejorar la clasificación y complementarla con un detector de falsos positivos pues enviar un comando erroneo al dron puede ser catastrófico. Tambien se considera trabajar con el reconocimiento de algun otro tipo de expresión tales como muecas, pestañeos e incluso pensamientos. Del mismo modo, se contempla experimentar con diferentes estímulos visuales que generen una reacción de alerta en el piloto tales como el observar que el dron se encuentre en riesgo de chocar. Finalmente, también se realizaran pruebas en escenarios menos controlados, como por ejemplo, realizar la detección y el control del dron en ambientes exteriores.

Agradecimientos. Este trabajo fue financiado por la Royal Society-Newton Advanced Fellowship con referencia NA140454. El autor Eduardo Zecua Corichi agradece el apoyo recibido por parte de CONACYT bajo la beca número 624062. El autor Irving Caballero Ledesma agradece el apoyo recibido por parte de CONACYT bajo la beca número 702771.

Referencias

1. Başar-Eroglu, C., Strüber, D., Schürmann, M., Stadler, M., Başar, E.: Gamma-band responses in the brain: a short review of psychophysiological correlates and functional significance. International Journal of Psychophysiology 24(1), 101–112 (1996)
2. Buzsaki, G.: Rhythms of the Brain. Oxford University Press (2006)
3. Corbetta, M., Miezin, F.M., Dobmeyer, S., Shulman, G.L., Petersen, S.E.: Attentional modulation of neural processing of shape, color, and velocity in humans. Science 248(4962), 1556 (1990)
4. Croft, R., Barry, R.: Removal of ocular artifact from the eeg: a review. Neurophysiologie Clinique/Clinical Neurophysiology 30(1), 5–19 (2000)
5. Gevins, A., Smith, M.E., McEvoy, L., Yu, D.: High-resolution eeg mapping of cortical activation related to working memory: effects of task difficulty, type of processing, and practice. Cerebral cortex 7(4), 374–385 (1997)
6. Harmony, T., Fernández, T., Silva, J., Bernal, J., Díaz-Comas, L., Reyes, A., Marosi, E., Rodríguez, M., Rodríguez, M.: Eeg delta activity: an indicator of attention to internal processing during performance of mental tasks. International journal of psychophysiology 24(1), 161–171 (1996)
7. Herrmann, C.S.: Human eeg responses to 1–100 hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. Experimental brain research 137(3-4), 346–353 (2001)
8. Huang, R.S., Jung, T.P., Delorme, A., Makeig, S.: Tonic and phasic electroencephalographic dynamics during continuous compensatory tracking. NeuroImage 39(4), 1896–1909 (2008)
9. Klimesch, W., Freunberger, R., Sauseng, P., Gruber, W.: A short review of slow phase synchronization and memory: evidence for control processes in different memory systems? Brain research 1235, 31–44 (2008)
10. Makeig, S., Delorme, A., Westerfield, M., Jung, T.P., Townsend, J., Courchesne, E., Sejnowski, T.J.: Electroencephalographic brain dynamics following manually responded visual targets. PLoS Biol 2(6), e176 (2004)
11. Makeig, S., Jung, T.P.: Tonic, phasic, and transient eeg correlates of auditory awareness in drowsiness. Cognitive Brain Research 4(1), 15–25 (1996)
12. Parasuraman, R., Cosenzo, K.A., De Visser, E.: Adaptive automation for human supervision of multiple uninhabited vehicles: Effects on change detection, situation awareness, and mental workload. Military Psychology 21(2), 270 (2009)
13. Picton, T.: Human brain electrophysiology. evoked potentials and evoked magnetic fields in science and medicine. Journal of Clinical Neurophysiology 7(3), 450–451 (1990)
14. Roberts, D.M., Taylor, B.A., Barrow, J.H., Robertson, G., Buzzell, G., Sibley, C., Cole, A., Coyne, J.T., Baldwin, C.L.: Eeg spectral analysis of workload for a part-task uav simulation. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. vol. 54, pp. 200–204. SAGE Publications (2010)
15. Valentino, D.A., Arruda, J., Gold, S.: Comparison of qeeg and response accuracy in good vs poorer performers during a vigilance task. International Journal of Psychophysiology 15(2), 123–133 (1993)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
septiembre de 2016
Printing 500 / Edición 500 ejemplares