

## SEGMENTACIÓN DE IMÁGENES UTILIZANDO CAMPOS ALEATORIOS DE MARKOV

### IMAGE SEGMENTATION USING MARKOV RANDOM FIELDS

JAIME FRANCISCO PANTOJA BENAVIDES<sup>1</sup>  
FRANK NIXON GIRALDO RAMOS<sup>2</sup>  
YEISSON STIVEN RUBIO VALDERRAMA<sup>3</sup>  
VÍCTOR MILCIADES ROJAS LARA<sup>4</sup>

RECIBIDO: ABRIL 2010  
APROBADO: AGOSTO 2010

#### RESUMEN

la segmentación es un proceso utilizado en visión artificial que consiste en dividir una escena en un conjunto de regiones, facilitando con ello la tarea de interpretación de una imagen. Los algoritmos de segmentación se basan en criterios como homogeneidad de la región y discontinuidad entre regiones disjuntas adyacentes. El presente artículo describe e implementa un modelo de segmentación que usa procesos de decisión estocásticos, el cual requiere apoyo y tiempo de computación para conseguir etiquetas óptimas, pero, entre otras ventajas, tiende a ser local y conduce a una implementación en *hardware* paralelo de manera natural.

#### Palabras clave

segmentación, visión artificial, campos aleatorios de Markov (CAM), vecindad, píxel, borde

#### Abstract

Segmentation is a process used in machine vision is to divide a scene into a set of regions, facilitating the task of interpretation of an image. Segmentation algorithms are

based on criteria such as homogeneity of the region and discontinuity between adjacent disjoint regions. This article describes and implements a segmentation model using stochastic decision processes, which requires the support and computing time to achieve optimal cloud, but, among other advantages, it tends to be local and leads to a parallel hardware implementation naturally.

1. Lic. en Matemáticas. Especialista en Estadística. Docente de la Universidad Distrital Francisco José de Caldas. Profesor investigador del grupo Sistemas Digitales Inteligentes (DIGITI). Correo electrónico: jfpantojab@udistrital.edu.co

2. Ingeniero en Control Electrónico e Instrumentación, Universidad Distrital Francisco José de Caldas. Investigador-director del grupo de investigación DIGITI. Correo electrónico: fngiraldor@udistrital.edu.co

3. Tecnólogo en Electrónica, Universidad Distrital Francisco José de Caldas. Investigador integrante del grupo DIGITI. Correo: rv3@hotmail.com

4. Tecnólogo en Electrónica, Universidad Distrital Francisco José de Caldas. Investigador integrante del grupo DIGITI. Correo electrónico: vmrl765@hotmail.com

**Key words:**

segmentation, artificial vision, Markov random fields (MRF), neighborhood, pixel, edge

**1. INTRODUCCIÓN**

La segmentación es un proceso importante en visión artificial, ya que divide una escena en un conjunto de regiones disjuntas, basándose para ello en características de similitud [1]. Es el primer nivel en la tarea de interpretación de la imagen y afecta significativamente procesos posteriores, proporcionando estructuras útiles tales como regiones y bordes.

Los algoritmos de segmentación generalmente están basados en la consideración de dos criterios: la homogeneidad de la región y la discontinuidad entre regiones disjuntas adyacentes [2]. Los campos aleatorios de Markov (CAM) tienen una potente capacidad para integrar varias informaciones visuales; han sido aplicados a problemas de visión tales como restauración y segmentación de imágenes [3]. El presente documento se organiza de la siguiente manera: transformaciones sobre imágenes, CAM aplicados a la segmentación, sistemas neuronales artificiales, algoritmo CAM, implementación de red neuronal, análisis de resultados y conclusiones.

**2. TRANSFORMACIONES****PRÁCTICAS REALIZADAS A UNA IMAGEN**

El procesamiento de datos en el sistema de visión puede enfocarse desde dos puntos de vista, a saber:

Alteración píxel a píxel de los datos en una escala global (operaciones individuales).

Operaciones basadas en múltiples puntos (transformaciones de vecindad).

**2.1. OPERACIONES INDIVIDUALES**

El proceso consiste en obtener el valor del píxel de una localización dada en la imagen, modificarlo con una operación lineal o no lineal, y colocar el valor del nuevo píxel en la correspondiente localización de la nueva imagen. Conocido como *operador umbral*, crea una imagen de salida binaria a partir de una escala de grises, donde el nivel de transición está dado por el parámetro de entrada  $P_1$ . La función de transformación es la siguiente:

$$q = \begin{cases} 0 & \text{si } p \leq p_1 \\ 255 & \text{si } p > p_1 \end{cases} \quad (1)$$

**2.2. TRANSFORMACIONES DE VECINDAD**

Son las transformaciones para modificar una imagen de entrada, bien en otra imagen distinta, o bien en una matriz de valores transformados, que no constituyen realmente una imagen, pero contienen la información necesaria que se espera de la transformación. Se dice que todo píxel  $p$  de coordenadas  $(x,y)$ , de vecindad-4 tiene cuatro píxeles que establecen con él la siguiente relación: horizontal:  $(x-1, y)$  y  $(x+1, y)$ , y vertical:  $(x, y-1)$  y  $(x, y+1)$ ; naturalmente, existen las excepciones dadas cuando el píxel  $(x, y)$  es un punto del frontera o esquina de la imagen, en cuyo caso algunos de los vecinos definidos anteriormente no existen.

### 2.2.1. OPERACIONES DE VECINDAD

Consisten, básicamente, en transformar el valor de un píxel  $p$  en la posición  $(x, y)$ , teniendo en cuenta los valores de los píxeles vecinos. Por ejemplo, si se considera una vecindad  $E_s(p)$ , se realiza una suma ponderada con los valores de los ocho vecinos, y el resultado de dicha suma es el valor del nuevo píxel  $q$  de la imagen de salida en la misma posición  $(x, y)$ . Por último, se establecen los valores de ponderación, definiendo una máscara con valores constantes. Dicha máscara es realmente un filtro, por lo que, dependiendo de su naturaleza, así será el resultado final. Por ejemplo, si se define la siguiente máscara:

$$\begin{bmatrix} 1,0 & 2,0 & 1,0 \\ 0,0 & 1,2 & 0,0 \\ -1,0 & -2,0 & -1,0 \end{bmatrix}$$

El valor del píxel  $q(x, y)$  vendría dado por la siguiente suma ponderada, con los factores de ponderación definidos por la máscara:

$$\begin{aligned} q(x, y) = & 1 * p(x-1, y-1) + 2 * p(x, y-1) + \\ & 1 * p(x+1, y-1) + 0 * p(x-1, y) + 1, 2 * p(x, y) + \\ & 0 * p(x+1, y) - 1 * p(x-1, y+1) - 2 * p(x, y+1) - \\ & 1 * p(x+1, y+1) \end{aligned} \quad (2)$$

Realizando esta transformación sobre todos los píxeles de la imagen original con la misma máscara, se obtendrá una nueva imagen de salida cuya dimensión es inferior a la original, [4].

La figura 2(b) muestra el resultado de aplicar la operación dada en máscara (2) sobre la imagen original de figura 2(a), con la máscara dada (2). Como se puede apreciar, el resultado es una especie de repujado en

relieve donde se marcan los bordes respecto del resto de la imagen, [5].

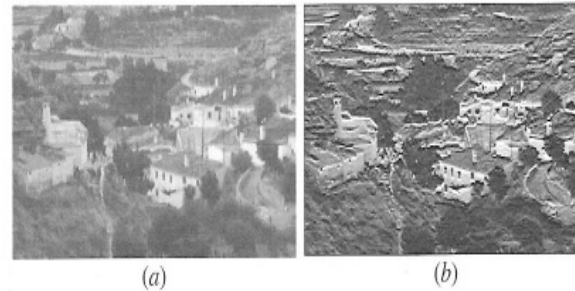


Figura 1: (a) imagen original; 2(b) Aplicación de la máscara (2) sobre la imagen (a).

### 3. CAMPOS ALEATORIOS DE MARKOV (CAM) PARA SEGMENTACIÓN DE IMÁGENES

Los CAM se fundamentan en procesos de decisión estocásticos con aplicaciones en un gran número de disciplinas del conocimiento, en especial en medicina, biomedicina y robótica; proporcionan una manera conveniente y constante de modelar entidades dependientes del contexto, tales como píxeles de la imagen y otras características espaciales correlacionadas; son atractivos por las siguientes razones:

- Hacen más fácil el cálculo de medidas de ejecución cuantitativas para caracterizar cómo trabajan los algoritmos de análisis de imagen.
- Los modelos de CAM pueden ser usados para incorporar información contextual a priori, o bien restricciones de manera cuantitativa.
- Los algoritmos basados en CAM tienden a ser locales y conducen a una implementación de hardware paralelo de manera natural.

La mayoría de los problemas en visión artificial se consideran de etiquetado, usando restricciones, debido al conocimiento a priori y a las observaciones. Los estudios basados en CAM han tenido éxito al modelar problemas de visión de bajo nivel, como restauración de imágenes, segmentación, reconstrucción de superficies, análisis de texturas, flujo óptico, integración visual y detección de bordes, [6].

### 3.1. EL PROBLEMA DEL ETIQUETADO

El problema de etiquetado consiste en asignar una etiqueta del conjunto de etiquetas  $L$  a cada estado de  $S$ ; por ejemplo, la detección de bordes en una imagen, que consiste en asignar una etiqueta  $f_i$  dentro del conjunto  $L = \{\text{borde, no borde}\}$  para cada estado  $i \in S$ , donde  $S$  está formado por los píxeles de la imagen. es decir,  $f = \{f_1, \dots, f_m\}$ , llamado *configuración* según CAM. En visión artificial, una configuración de etiquetas puede corresponder a una imagen, un mapa de bordes de la imagen, una interpretación de los objetos que aparecen en una escena, etc.

### 3.2. CAMPOS ALEATORIOS DE MARKOV (CAM) Y DISTRIBUCIÓN DE GIBBS (DG)

Los CAM son una parte de la teoría de probabilidad que proporciona una herramienta para analizar dependencias espaciales o contextuales de fenómenos físicos. Se usan en el problema del etiquetado en visión, para establecer las distribuciones de probabilidad de las etiquetas, [7].

#### 3.2.1. CAMPOS ALEATORIOS DE MARKOV (CAM)

Los modelos de CAM no son un método de segmentación en sí mismos, pero sí un mo-

delo estadístico que puede ser usado dentro de los métodos de segmentación. Los CAM modelan las interacciones espaciales entre vecinos o píxeles cercanos; estas correlaciones locales proveen un mecanismo para modelar una variedad de propiedades de la imagen, [8].

Sea  $F = \{f_1, \dots, f_m\}$  una familia de variables aleatorias definidas en  $S$ , donde cada variable aleatoria  $F_i$  toma un valor  $f_i$  en  $L$ . Se llamara campo aleatorio a la familia de variables aleatorias  $F$ . Se dice que la familia

$$\begin{aligned}
 P(F = f) &> 0 \\
 P(F_i = f_i / F_j = f_j, j \in d, j \neq i) &= P(F_i = f_i / F_j = f_j, j \in N_i)
 \end{aligned}
 \tag{3}$$

#### 3.2.2. SISTEMA DE VECINOS Y CLIQUES

##### 3.2.2.1. VECINDAD

Los estados de  $S$  están relacionados mediante lo que llamamos un sistema de vecinos  $N = \{N_i / \forall i \in S\}$ , donde  $N_i$  es el conjunto de los estados vecinos de  $i \in S$ .

Para una rejilla regular  $S$ , el conjunto de vecinos de  $i$  está definido como el conjunto de estados próximos dentro de un radio  $r$ , es decir,

$$N_i = \{j \in S / [dist(pixel_j, pixel_i)]^2 \leq r, j \neq i\} \tag{4}$$

donde  $dist(A, B)$  denota la distancia euclídea entre  $A$  y  $B$ , y  $r$  es un número entero.

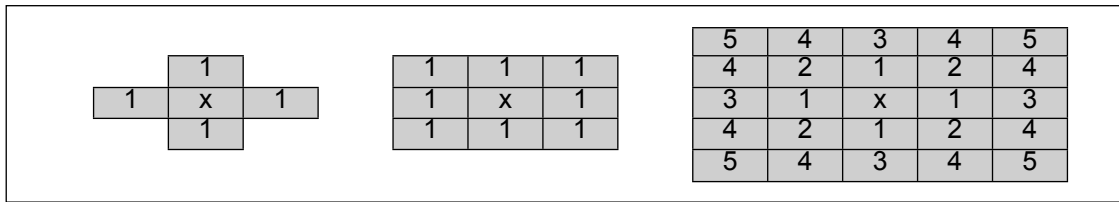


Figura 2. Vecinos en una rejilla regular S

Para cada estado interior, representado por  $(i,j)$ , de vecindad 4, está definido como:

$$N_{i,j} = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\} \tag{5}$$

Los estados de la frontera tienen tres y los estados de las esquinas tienen solamente dos, [9].

**3.2.2.2. CLIQUE**

Un clique  $C$  para  $\{S, N\}$  se define como un subconjunto de  $S$  tal que  $C$  consiste en un único estado  $C = \{i\}$  o un par de estados vecinos  $C = \{i,j\}$ , o tres estados vecinos  $C = \{i,j,k\}$ , y así sucesivamente.

El tipo de clique para  $\{S, N\}$  de una rejilla regular viene determinado por su tamaño, dimensión y orientación. La figura 3 muestra distintos tipos de cliques para sistemas de vecinos de primer y segundo orden. A medida que aumenta el orden del sistema de vecinos, el número de cliques crece rápidamente y también el coste computacional [10].

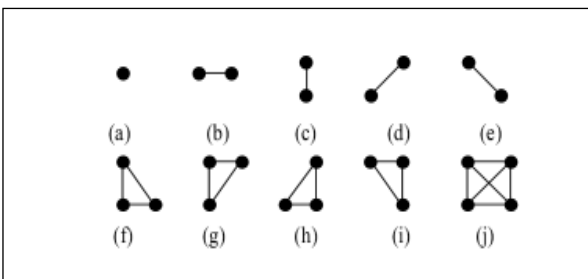


Figura 3. Cliques para vecindad 4 y 8

**4. SISTEMAS NEURONALES ARTIFICIALES**

En la teoría de la resonancia adaptativa (ART), los esquemas de aprendizaje competitivo no tienen garantía explícita sobre la estabilidad de los grupos de patrones de entrada durante el proceso de decrecimiento del factor de aprendizaje en las iteraciones sucesivas. Las redes basadas en la teoría de resonancia adaptativa vienen a solucionar parte de este problema, [11].

En este tipo de redes, los vectores de pesos del elemento de procesamiento seleccionado como ganador sólo se actualizan con los patrones de entrada que “resuenan” con estos, es decir, si son “suficientemente similares”; si no lo son, se genera un nuevo elemento de procesamiento cuyos pesos son precisamente los del patrón de entrada utilizado. El entrenamiento de la red es de tipo competitivo, esto es, de modo autónomo, la red encuentra los valores de pesos de conexiones y también es capaz de crear nuevos valores de ser necesario, [12].

Las variables  $w$  representan el valor del peso de cada conexión. Como todas las neuronas de la capa de entrada se conectan con todas las neuronas de la capa de salida, existe un valor de peso  $w$  para cada conexión. Los pesos  $w$  representan los pesos hacia adelante o *feedware*, y los pesos identificados con la letra  $v$  representan los pesos hacia atrás o *feedback*.

Las funciones de activación para las neuronas de salida  $Y_1$  y  $Y_2$  son:

$$Y_1 = f(x1*w11 + x2*w21 + xi*wi1 + \dots + xn*wn1)$$

$$Y_2 = f(x1*w12 + x2*w22 + xi*wi2 + \dots + xn*wn2)$$

(6)

donde la neurona de salida que dé como resultado el valor más alto será la neurona ganadora.

### 5. IMPLEMENTACIÓN DE OPERADORES Y ALGORITMO CAM PARA LA SEGMENTACIÓN DE IMÁGENES

Anteriormente se describieron operaciones para realizar el acondicionamiento de imágenes, con el fin de trabajar imágenes mejoradas (sin ruido) y con la mayor cantidad de información posible. Fue necesario realizar la implementación de cada uno de los algoritmos y las ecuaciones mostradas atrás, con

el ánimo de encontrar las fortalezas y las debilidades con respecto al algoritmo CAM.

En esta sección se aporta un modelo concreto de CAM para segmentación de imágenes, a través de la incorporación de distintas características a la función de energía que guía el proceso de segmentación. Este proceso divide una escena en un conjunto de regiones disjuntas, basándose para ello en características similares como intensidad, color o textura. La segmentación es uno de los elementos más importantes de cualquier sistema automatizado de visión; es el primer nivel de la tarea de entendimiento de la imagen y afecta considerablemente el proceso posterior de interpretación de la imagen, proporcionando estructuras útiles tales como regiones y bordes. A continuación se describen los operados de extracción de bordes más utilizados.

#### 5.1. OPERADOR ROBERTS

Corresponde a las diferencias cruzadas de 2 x 2. Las máscaras utilizadas en este operador son:

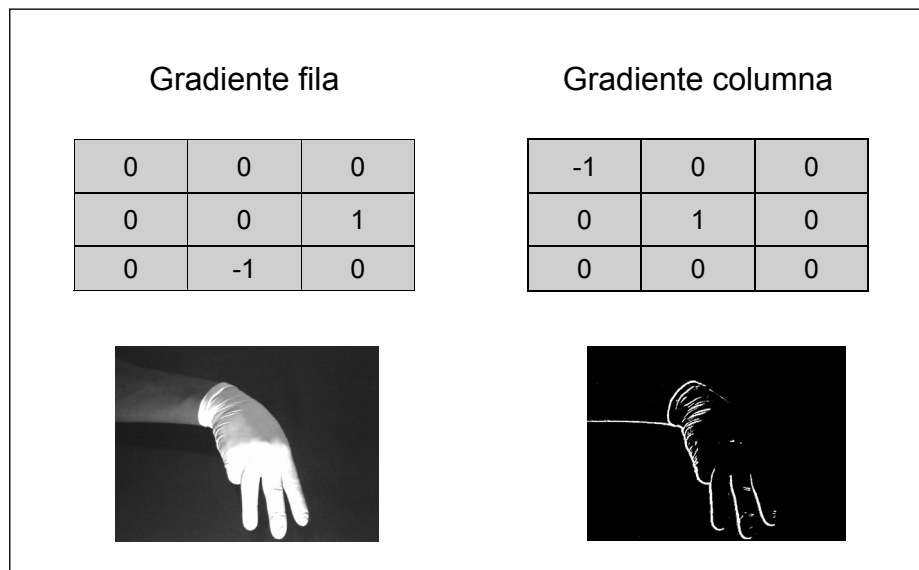


Figura 4. Resultados de la segmentación con el operador Roberts

**5.2. OPERADORES DE PREWITT, SOBEL Y FREI-CHEN**

Los tres operadores pueden formularse de forma conjunta con las siguientes máscaras de convolución:

Gradiente fila			Gradiente columna				
$\frac{1}{2+k}$	1	0	-1	$\frac{1}{2+k}$	-1	-k	-1
	k	0	-k		0	1	0
	1	0	-1		1	k	1

En el operador Prewitt ( $K=1$ ) se involucran a los vecinos de filas / columnas adyacentes para proporcionar mayor inmunidad al ruido.

de Prewitt, aunque en la práctica hay poca diferencia entre ellos.

El operador Sobel ( $K=2$ ) se supone que es más sensible a los bordes diagonales que el

En Frei-Chen ( $K=\sqrt{2}$ ), el gradiente es el mismo para bordes verticales, horizontales y diagonales.

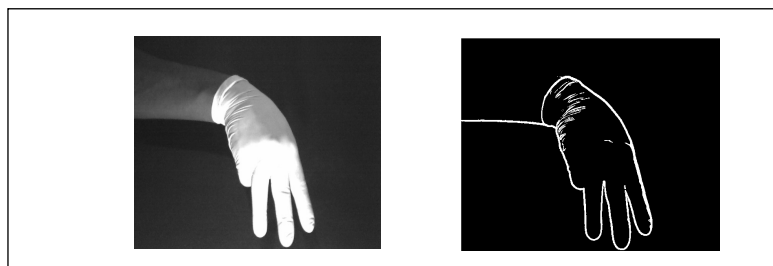


Figura 5. Resultados de la segmentación con el operador Prewitt

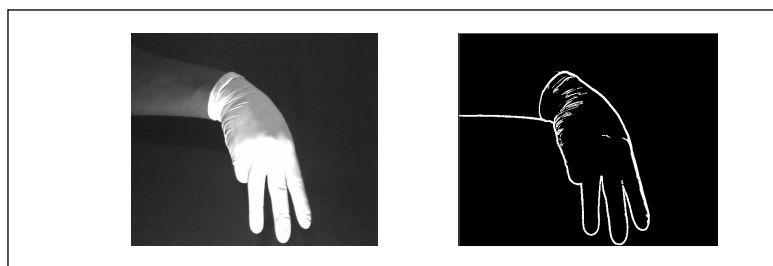


Figura 6. Resultados de la segmentación con el operador Frey-Chen

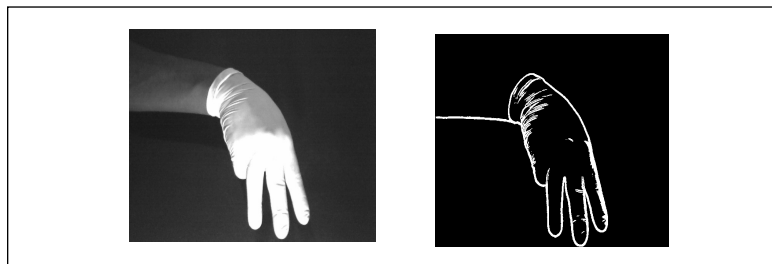


Figura 7. Resultados de la segmentación con el operador Sobel

### 5.3 SEGMENTACIÓN CON CAMPOS ALEATORIOS DE MARKOV

En CAM la segmentación se define como una función de energía del siguiente modo:

$$U = \sum_i C_i \quad (7)$$

dónde:  $C_i$  = cada uno de los cliques de vecindad 4. Además, cada función clique viene dada por:

$$C_i = \max\left(|\mu_i - \mu_j|, |\sigma_i - \sigma_j|\right) \quad (8)$$

donde  $\mu$  = media de intensidad de la region  $i$ , y  $\sigma$  = desviación estandar de la region  $i$ .

De las figuras 4 a la 7 se muestra la segmentación de imágenes con los operadores; se puede observar que estos no eliminan satisfactoriamente el ruido dentro de la imagen a segmentar. En la figura 8, por su parte, se presentan los resultados de la segmentación por CAM; se observa una segmentación con menos ruido, en comparación con los operadores clásicos.

### 6. IMPLEMENTACIÓN ALGORITMO RED NEURONAL

La arquitectura de red neuronal elegida para realizar el procedimiento de reconocimiento de imágenes fue la de tipo ART (Adaptive Resonance Theory). Una vez realizado el procesamiento previo de imágenes (transformaciones a escala de grises, eliminación de ruido y extracción de contornos), se hizo la elección de tres imágenes fácil-

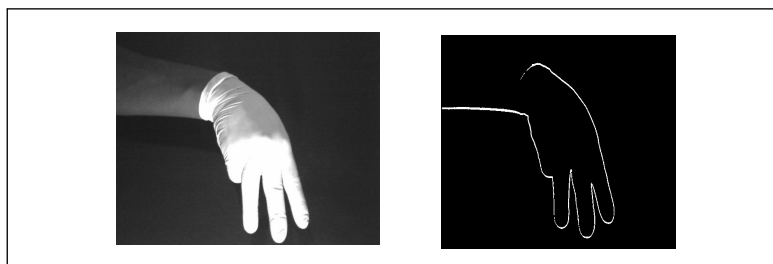


Figura 8. Resultados segmentación con campos aleatorios de Markov



mente diferenciables, para realizar el proceso de entrenamiento y procesamiento de la red neuronal.

El primer paso consistió en determinar el número de neuronas y capas para la configuración de la red neuronal. De este modo, se trabajó con tres neuronas de salida y una capa de entrada de 307.200 neuronas. Como las imágenes a trabajar tienen una resolución de 640x480 píxeles, es necesario dedicar una neurona para cada uno de estos; por tal razón, el número de neuronas de la capa de entrada es de  $307.200 = 640 \times 480$ .

La red neuronal debe procesar la imagen una vez el algoritmo haya hecho el proceso de captura, transformación a escala de grises, eliminación de ruido y extracción el contorno.

Para el procesamiento de la red, el primer paso consiste en transformar la imagen de su forma básica matricial (640x480 píxeles) a una forma vectorial, para poder designar una neurona a cada posición del vector, el cual tendrá el valor de cada uno de los píxeles de la imagen segmentada y binarizada.

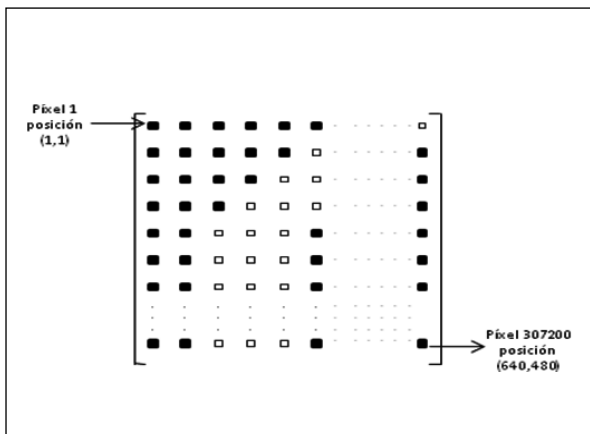


Figura 9a. Representación imagen en forma matricial de 640x480 píxeles

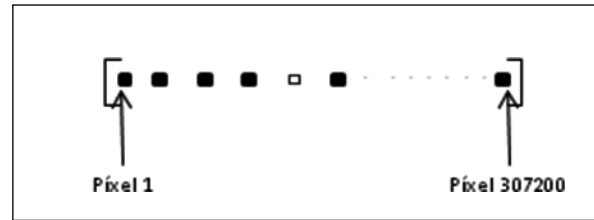


Figura 9b. Representación imagen en forma vectorial de 307.200 píxeles

En la figura 9a se muestra la representación de una imagen en forma matricial; claro está, la imagen ha sido binarizada previamente, es decir, los valores de cada píxel se encuentran entre 0 y 1, pero estos valores han sido representados en forma de cuadros blancos y negros, para una mejor ilustración. En la figura 9b se representa la imagen vectorizada.

En el proceso de entrenamiento de la red neuronal, el primer patrón que se aplica es una imagen vectorizada de la letra E. Como se trata de la primera información, se realiza el aprendizaje completo de esta, actualizándose los pesos de las conexiones para almacenar este ejemplar, que será el primer representante de cada categoría o patrón. Los valores de los pesos para las variables  $w$  fueron determinados de acuerdo con cada una de las imágenes patrones de las letras escogidas. El algoritmo de la red neuronal ART debe transformar la imagen a forma vectorial y realizar el producto entre el valor de cada píxel por su respectivo valor de peso  $w$ , activando para ello las neuronas de salida que correspondan.

Para este desarrollo se trabajó con las neuronas de salida llamadas *Letra\_E*, *Letra\_G*, *Letra\_O* y *Letra\_P*.

La red neuronal tiene una base de datos de 20 imágenes, cinco para cada patrón. Como

fue explicado antes, las redes neuronales tipo ART operan de modo competitivo; la red comprobará el grado de semejanza de la nueva imagen y los 20 prototipos almacenados en la red. Al ser información binaria, y suponiendo que un píxel negro se representa mediante un bit 1 en el vector de entrada, y uno blanco con un bit 0, se puede comprobar el grado de semejanza aplicando la siguiente función de activación:

$$\frac{E_K X}{E_K} \quad (9)$$

Donde las  $X$  representan la imagen de entrada ya vectorizada, y  $E_K$  cada uno de los patrones. En el numerador se indica el número de píxeles negros coincidentes (operación AND entre ambas figuras), y en el denominador, el número de píxeles negros del patrón; así se determina el nivel de semejanza entre la entrada y las imágenes de la base de datos.

El resultado de este cálculo determina cuál de las neuronas (*Letra\_E*, *letra\_G*, *letra\_O* y *Letra\_P*.) es la ganadora (la de mayor valor). El resultado de la red neuronal indica cuál es la imagen más parecida a la que el usuario está realizando. El *software*, por su parte, solamente debe visualizar en la pantalla la letra reconocida.

## 6.1. ENTRENAMIENTO DE LA RED NEURONAL

El entrenamiento de la red neuronal se realizó en el programa MATLAB®, donde se implementó un algoritmo para realizar el cálculo de los valores de los pesos para las conexiones de las neuronas.

En el proceso de entrenamiento de la red neuronal se consideraron cuatro patrones claramente diferenciables entre sí, los cuales se tomaron como referencia para el cálculo de los pesos de la red.

En la operación del algoritmo de tratamiento y procesamiento de la imagen, se aplicó una transformación de esta, de su formato matricial a un formato vectorial, pero con la diferencia de que esto se hace para cada una de las imágenes patrones.

## 7. ANÁLISIS DE RESULTADOS

Se verificó el método de segmentación de imágenes con campos aleatorios de Markov y se realizó una comparación con los operadores comúnmente usados. Se logró comprobar el algoritmo en operación sobre imágenes en un ambiente controlado, donde se redujo radicalmente la cantidad de ruido y los bordes de los objetos a reconocer quedaron con una mayor definición.

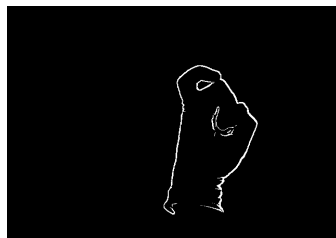


Figura 10. CAM

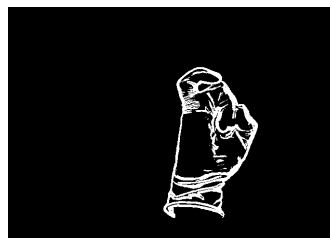


Figura 11. Sobel

En las figura 10 y 11 se puede observar claramente cómo empleando los campos aleatorios de Markov, la incidencia del ruido es menor y se puede efectuar una mejor extracción de contornos, en comparación con otros operadores convencionales, en este caso el operador Sobel.

## 8. CONCLUSIONES

Aplicando la función de energía se mide la intensidad de las regiones. En el modelo de CAM, el promedio realiza una segmentación de las imágenes con menor ruido que aquel que se presenta con los operadores comúnmente utilizados, lo que en definitiva le permitirá a la red neuronal funcionar y reconocer con mayor exactitud.

Se definió un operador de vecindad que permitió distinguir propiedades de individualidad; es decir, si la red neuronal está formada por cúmulos de píxeles, o por píxeles aislados, y con base en esto, determinar cuáles píxeles son ruido y normalizarlos.

Se presentó un estudio sobre el desempeño de técnicas de clasificación de patrones para el reconocimiento de gestos manuales. Las técnicas estudiadas abarcan un amplio espectro y parámetros de inteligencia artificial.

## REFERENCIAS

- [1] O. Faugeras, *Three-Dimensional Computer Vision A Geometric Point of View*. Cambridge: The Mit Press, 1996.
- [2] R. M. Haralick, *Computer and Robot Vision*, vol. 2. Reading: Addison-Wesley, 1993.
- [3] S. Geman y D. Geman, "Stochastic relaxation, Gibbs Distribution and the Bayesian Restoration of Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721-41, 1984.
- [4] R. Olivares, *Procesamiento digital de imágenes y visión*. Consultado en mayo de 2009 en [http://antares.itmorelia.edu.mx/~jcolivar/courses/ag08a/ag\\_u3.ppt](http://antares.itmorelia.edu.mx/~jcolivar/courses/ag08a/ag_u3.ppt), 2007.
- [5] Cortes B., Oscar J. Reconocimiento del lenguaje de gestos manuales alfabéticos mediante visión artificial, Ingeniería en Control e Instrumentación Electrónica, UNIVERSIDAD DISTRITAL "Francisco José de Caldas" Facultad Tecnológica, 2007.
- [6] M. Pujol, R. Rizo, P. Arques, P. Compañy, F. Escolano, y R. Molina, "Aplicación de los modelos de campos aleatorios de Markov en visión artificial", *Revista Electrónica de Visión por Computador*, no. 4, Departamento de Ciencia de la Computación e Inteligencia Artificial F. Pujol, Departamento de Tecnología Informática y Computación, Universidad de Alicante, octubre de 2000.
- [7] E. López, "Segmentación de coberturas de la tierra espectralmente similares empleando campos aleatorios de Markov, características y geometría estocástica", consultado en octubre de 2009 en <http://ccc.inaoep.mx/~danae/PropuestaDoctoral.pdf>.
- [8] M. Fernández, "Campos aleatorios de Markov", consultado en junio de 2009 en <http://lmi.bwh.harvard.edu/papers/>

- pdfs/2004/martinfernandezCOURSE04c.pdf, 2004.
- [9] F. Fernández y D. Borrajo, “Ingeniería informática, aprendizaje automático”, Grupo de Planificación y Aprendizaje (PLG), Departamento de Informática, Escuela Politécnica Superior, Universidad Carlos III de Madrid, consultado en julio de 2009 en <http://www.plg.inf.uc3m.es/~aa/transpas08-09/presentacion-08-09.pdf>, 2009.
- [10] L. Sucar. PGM - CAM, “Campos de Markov, sesion 6”, consultado en septiembre de 2009 en <http://ccc.inaoep.mx/~esucar/Clases-incer/clase07-cm.ppt>.
- [11] J. Gúmez, J. Peñaloza, J. Rugeles, y F. Caicedo, “Sistema neuronal de bajo costo para la identificación de colores”, Grupo de Investigaciones en Procesamiento de señales (GSP), Universitaria de Investigación y Desarrollo (UDI), Grupo de Percepción y Sistemas inteligentes, Universidad del Valle. Consultado en septiembre de 2009 en: [http://docs.google.com/viewer?a=v&q=cache:pBw-2dW8eRUJ:gps.udi.edu.co/archivos/publicaciones/SistemaNeuronaldeBajoCostoparalaidentificaciondecolores.pdf+Sistema+Neuronal+de+Bajo+Costo+para+la+Identificaci%C3%B3n+de+Colores&hl=es&gl=co&pid=bl&srcid=ADGEEsIuXpWlnastPHObVSYivQBKbTPZZOVRebJ4IpKm3kfhT96OAJAtwbN85GqJBc9fzMadFiWiRZzTOB9J0VfvTOSWYEAuYq1kSNiOb2\\_0nN9T\\_J\\_F3M4-px80AmusEk1YpKk1yvXE&sig=AHIEtbSb8LugwxGMxvUTs7QenS0IpppEaw](http://docs.google.com/viewer?a=v&q=cache:pBw-2dW8eRUJ:gps.udi.edu.co/archivos/publicaciones/SistemaNeuronaldeBajoCostoparalaidentificaciondecolores.pdf+Sistema+Neuronal+de+Bajo+Costo+para+la+Identificaci%C3%B3n+de+Colores&hl=es&gl=co&pid=bl&srcid=ADGEEsIuXpWlnastPHObVSYivQBKbTPZZOVRebJ4IpKm3kfhT96OAJAtwbN85GqJBc9fzMadFiWiRZzTOB9J0VfvTOSWYEAuYq1kSNiOb2_0nN9T_J_F3M4-px80AmusEk1YpKk1yvXE&sig=AHIEtbSb8LugwxGMxvUTs7QenS0IpppEaw)
- [12] S.J. Russell y P. Pórvlg, *Inteligencia artificial, un enfoque moderno*, 2ª ed. Madrid: Pearson Educación, 2004.