

Transformada para imágenes basada en memorias asociativas Alfa-Beta

Antonio Alarcón Paredes, Oleksiy Pogrebnyak y Amadeo José Argüelles Cruz

Centro de Investigación en Computación, Instituto Politécnico Nacional,
México

aparedesb07@sagitario.cic.ipn.mx, {olek, jamadeo}@cic.ipn.mx

Resumen. En este artículo se presenta un método de memorias asociativas alfa-beta para imágenes (MAABI), que origina la transformada alfa-beta (TAB) para imágenes. Esta transformada se aplica a sub-bloques de una imagen en escala de grises y genera una memoria heteroasociativa alfa-beta en cada sub-bloque, a partir de una matriz de transformación dada con la transformada alfa-beta inversa (TABI) recuperamos los patrones de la imagen original. El proceso de compresión de datos se divide en tres fases: transformación, cuantificación y codificación; este modelo se enfoca en la transformación de la imagen para compararse con la transformada morfológica (TM), basada en las memorias asociativas morfológicas (MAM). Se utiliza la entropía de Shannon como medida de la cantidad de bits contenidos en cada imagen. La TM, al igual que métodos tradicionales, como la transformada discreta del coseno (DCT) o la transformada discreta wavelet (DWT), no ofrece compresión alguna de la imagen; por lo que una de las ventajas ofrecidas por la TAB es que la imagen transformada posee una menor entropía que la imagen original. Asimismo, la TAB ofrece rapidez en el procesamiento con un reducido número de operaciones elementales, tales como sumas, restas, máximos y mínimos.

Palabras clave. Métodos de transformadas, reconocimiento de patrones, procesamiento de imágenes, memorias asociativas, compresión.

Image Transform based on Alpha-Beta Associative Memories

Abstract. In this paper, a new method of alpha-beta associative memories for images (MAABI) is

presented. This method results in the alpha-beta transform (TAB) for images. The alpha-beta transform presented in this paper is applied to sub-blocks of a gray-scale image and generates an alpha-beta heteroassociative memory on each sub-block using a given transformation matrix. By means of the inverse alpha-beta transform (TABI), the original image patterns are recovered. The data compression process is divided into three stages: transformation, quantization and coding. Our model is focused on image transformation; we compare it with the morphological transform (TM) based on the morphological associative memories (MAM). In order to measure the number of bits contained in an image, the Shannon's entropy is used. The TM, as well as such traditional transforming methods as the discrete cosine transform (DCT) and the discrete wavelet transform (DWT), cannot perform compression over an image; that is why the main advantage of the AB is a smaller value of the entropy measure of the transformed image compared to the entropy of the original image. In addition, TAB provides a faster processing with a small number of elementary operations such as addition, subtraction, maximum and minimum.

Keywords. Transform methods, pattern recognition, image processing, associative memories, compression.

1 Introducción

Dentro de un sistema de compresión de imágenes, la fase de transformación juega un papel importante. Esta etapa no provee compresión de información, su finalidad radica en

reducir la dependencia entre los píxeles de una imagen ayudando y facilitando la tarea de compresión de la imagen en las fases posteriores. En este artículo se presentan las memorias asociativas alfa-beta para imágenes (MAABI) que dan origen a la transformada alfa-beta (TAB). Este método se enfoca en la etapa de transformación de la imagen y se compara con la transformada morfológica (TM), Guzmán *et al.* [6], basada en las memorias asociativas morfológicas (MAM), Ritter *et al.* [14]. Aun cuando existe un gran número de transformadas que pueden utilizarse en la compresión de imágenes, lo común es que se utilicen métodos tradicionales, como la transformada discreta del coseno (DCT, *discrete cosine transform*) o la transformada discreta wavelet (DWT, *discrete wavelet transform*).

La DCT fue propuesta por Ahmed, Natarajan y Rao en 1974 [1]. Dos años más tarde Haralick, [8] muestra que la DCT de N puntos puede ser calculada con mayor rapidez tomando dos transformadas rápidas de Fourier (FFT) de N puntos en lugar de tomar una FFT de $2N$ puntos. A partir de esto, los esfuerzos por obtener una menor complejidad computacional en el algoritmo original de la DCT no han cesado. En [4] se propone la implementación de una DCT rápida donde se toma en consideración la simetría de la función coseno, reduciendo la cantidad de operaciones necesarias para el cálculo de esta transformada. Once años más tarde, se propuso un esquema rápido de la DCT aplicada a imágenes [3] donde utilizan la teoría de Haralick, [8] tomando sólo la parte real de la transformada discreta de Fourier (DFT) como se menciona en Tseng y Miller [20].

A través de los años, se han estandarizado varios algoritmos de compresión de imágenes que utilizan la DCT; tal es el caso del estándar de compresión para imágenes fijas JPEG [21] y del estándar para imágenes en video MPEG [MPEG-2video, 1995].

Posteriormente, Daubechies plasma la teoría y fundamentos matemáticos de la Transformada Wavelet [5]. Partiendo de esta teoría Antonini, junto a la misma Daubechies y otros colaboradores [2] utilizan filtros biortogonales y descomponen la wavelet en varios niveles proveyendo imágenes sub-banda que

corresponden a diferentes niveles de resolución y orientación de la wavelet.

Se han generado varios métodos de compresión utilizando la DWT, de entre los que destacan el *set partitioning in hierarchical trees* (SPIHT) [15], *set partitioning embedded block* (SPECK) [10] y el *embedded zerotrees of wavelet transforms* (EZT) [17], siendo este último uno de los más utilizados.

Para el año 2000, Taubman propone un nuevo algoritmo de compresión de imágenes conocido como *embedded block coding with optimized truncation* (EBCOT) [19], consiste en que cada sub-banda se divide en pequeños bloques de coeficientes wavelet, denominados bloques de código (*blocks code*) y genera cadenas de bits para cada bloque. El algoritmo EBCOT en conjunto con la DWT son la base fundamental del estándar de compresión de imágenes JPEG2000 [11].

La TM fue propuesta como un método alternativo a las transformadas tradicionales mencionadas con anterioridad. Esta transformada genera una representación de la imagen basada en operaciones que calculan máximos y mínimos de sumas o restas. En Guzmán [7] se describe a la TM como un método atractivo para facilitar la compresión en sus etapas siguientes, ya que es reversible, posee bajos requerimientos de memoria y utiliza precisión aritmética limitada.

La TM se aplica a los sub-bloques individuales de la imagen, que junto con una matriz de transformación constituyen los datos de entrada de las MAM. Podemos decir que la matriz de transformación utilizada en la TM es una matriz identidad, con la salvedad de que en su diagonal principal contiene el valor máximo que puede encontrarse en un píxel de la imagen. Así, la TM provee una imagen resultante donde en cada uno de sus sub-bloques se generó una MAM. Los valores contenidos en cada MAM generada son el resultado de obtener el máximo, o mínimo, de una resta entre la matriz de transformación y su respectivo valor de píxel en cada sub-bloque. La TM parte de la idea que al reorganizar y escalar los valores de cada sub-bloque, se disminuya la relación entre los píxeles de la imagen, reduciendo así la redundancia espacial. La TM, al igual que la DCT y la DWT no ofrece compresión, por lo que la ventaja de la TM sobre las dos

anteriores es la sencillez y rapidez de sus operaciones. Los detalles del completo funcionamiento de la TM pueden ser consultados en Guzmán [7].

El artículo está organizado de la siguiente manera: en la Sección 2 se muestra la teoría de las MAABI, mientras que en la Sección 3 se describe el algoritmo de la TAB. Los resultados y las conclusiones se mostrarán en la Sección 4 y la Sección 5, respectivamente.

2 Teoría de las memorias asociativas Alfa-Beta para imágenes

Las memorias asociativas (MA) [9] son algoritmos de reconocimiento de patrones, cuyo propósito es recuperar patrones completos a partir de patrones de entrada que pueden presentar alteraciones aditivas, sustractivas o mezcladas.

Los patrones de entrada son representados por vectores columna x de dimensión n y los patrones de salida por vectores columna y de dimensión m . Cada patrón de entrada x se corresponde con uno y sólo un patrón de salida y formando una asociación a modo de un par ordenado: (x,y) .

El conjunto de las p asociaciones de patrones $\{(x^1, y^1), (x^2, y^2), \dots, (x^p, y^p)\}$ se denomina *conjunto fundamental de asociaciones* o simplemente *conjunto fundamental*, y está representado como:

$$\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\} \quad (1)$$

Las MA para su funcionamiento se dividen en dos fases: la fase de aprendizaje, donde se asocian los patrones de entrada x con su correspondiente patrón de salida y para generar la memoria M , como se ejemplifica en la Fig. 1; y la fase de recuperación, en la cual presentamos un patrón de entrada x a la memoria M y como resultado esperamos que nos arroje su correspondiente patrón de salida y y cómo podemos observar en la Fig. 2.

Existen dos tipos de MA que se clasifican de acuerdo con la naturaleza de sus patrones: MA autoasociativas y MA heteroasociativas. Una memoria es autoasociativa si cumple que



Fig. 1. Fase de aprendizaje de una MA



Fig. 2. Fase de recuperación de una MA

$\forall \mu, x^\mu = y^\mu$, es decir, cada patrón de entrada es igual a su correspondiente patrón de salida; por otra parte, la memoria será heteroasociativa si se cumple que $\exists \mu \in \{1, 2, \dots, p\} \mid x^\mu \neq y^\mu$, es decir, si existe al menos un patrón de entrada diferente a su correspondiente patrón de salida.

Para el caso de las MAM y las MAABI, ya sean autoasociativas o heteroasociativas, cada una de estas puede subdividirse en MA tipo *máx*, que denotaremos como M , o MA tipo *mín*, que denotaremos como W [23]. En este trabajo se utilizan las MAABI en su modo heteroasociativo, debido a las características que se requieren para la transformación de la imagen.

2.1 Funcionamiento de las memorias asociativas Alfa-Beta para imágenes

Las memorias asociativas alfa-beta [24] utilizan máximos y mínimos, y dos operaciones originales α y β de las cuales heredan el nombre. Para la definición de las operaciones α y β se deben especificar los conjuntos A y B, los cuales son: $A = \{0, 1\}$ y $B = \{0, 1, 2\}$. Como las componentes de los patrones contienen valores en el conjunto A, estas memorias funcionan para entradas y salidas con valores binarios. Los valores dentro de la memoria generada pertenecen al conjunto B. Debido a que las imágenes que vamos a tratar son de 8 bits en escala de grises y su rango de valores va de 0 a 255, fue necesaria una modificación a los operadores Alfa y Beta, con la finalidad de extenderlos al campo de los números reales.

La nueva operación Alfa para valores reales $\alpha_{\mathbb{R}} = \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ está definida como:

$$\alpha_{\mathbb{R}}(c, d) = c - d + 1 \quad (2)$$

Para el caso de la operación $\beta_{\mathbb{R}} = \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, tenemos una operación para cada tipo de memoria, máx o mín, quedando de la siguiente manera:

$$\text{Si } c = d \rightarrow \beta(c, d) = c \quad (3)$$

$$\beta_{\mathbb{R}}^{\vee}(c, d) = d - |c| - 1 \quad (4)$$

$$\beta_{\mathbb{R}}^{\wedge}(c, d) = c - |d| - 1 \quad (5)$$

Habiendo definido estas operaciones, la notación para las operaciones matriciales queda como sigue:

Sea el vector columna \mathbf{y} de dimensión m y el vector fila \mathbf{x} de dimensión n , la operación $\alpha_{\mathbb{R}}$ entre éstos está definida como

$$\mathbf{y} \otimes_{\mathbb{R}} \mathbf{x} = [f_{ij}]_{m \times n} \quad (6)$$

donde

$$f_{ij} = \alpha_{\mathbb{R}}(y_i, x_j) = y_i - x_j + 1 \quad (7)$$

con $i=1, 2, \dots, m$ y $j=1, 2, \dots, n$.

Sea $\mathbf{Q} = [q_{ij}]_{m \times n}$ una matriz y sea $\mathbf{x} = [x_i]_n$ un vector columna. La operación $\beta_{\mathbb{R}}^{\vee}$ está definida como

$$\mathbf{Q} \nabla_{\beta_{\mathbb{R}}} \mathbf{x} = [y_i]_m \quad (8)$$

donde

$$y_i = \bigvee_{j=1}^n \beta_{\mathbb{R}}^{\vee}(Q_{ij}, x_j) \quad (9)$$

Sea $\mathbf{Q} = [q_{ij}]_{m \times n}$ una matriz y sea $\mathbf{x} = [x_i]_n$ un vector columna. La operación $\beta_{\mathbb{R}}^{\wedge}$ está definida como

$$\mathbf{Q} \Delta_{\beta_{\mathbb{R}}} \mathbf{x} = [y_i]_m \quad (10)$$

donde

$$y_i = \bigwedge_{j=1}^n \beta_{\mathbb{R}}^{\wedge}(Q_{ij}, x_j) \quad (11)$$

2.1.1 Memorias heteroasociativas Alfa-Beta mín

La fase de aprendizaje consta de dos pasos; en el primero se utiliza el operador $\otimes_{\mathbb{R}}$, mientras que en el segundo paso debe aplicarse el operador mínimo \bigwedge . 1. Para cada $\mu = \{1, 2, \dots, p\}$ a partir de la pareja $(\mathbf{x}^{\mu}, \mathbf{y}^{\mu})$ se construye la matriz $\mathbf{y}^{\mu} \otimes_{\mathbb{R}} (\mathbf{x}^{\mu})^t$. 2 Se aplica el operador mínimo a las matrices obtenidas en el paso 1 para obtener la memoria \mathbf{W} .

$$\mathbf{W} = \bigwedge_{\mu=1}^p [\mathbf{y}^{\mu} \otimes_{\mathbb{R}} (\mathbf{x}^{\mu})^t] = w_{ij} \quad (12)$$

$$w_{ij} = \bigwedge_{\mu=1}^p \alpha_{\mathbb{R}}(y_i^{\mu}, x_j^{\mu})$$

En la fase de recuperación se presenta un patrón \mathbf{x}^{ω} , con $\omega \in \{1, 2, \dots, p\}$, a la memoria heteroasociativa tipo *mín* y se realiza la operación $\mathbf{W} \nabla_{\beta_{\mathbb{R}}} \mathbf{x}^{\omega}$ para obtener su correspondiente patrón de salida \mathbf{y}^{ω} :

$$y_i^{\omega} = \bigvee_{j=1}^n \beta_{\mathbb{R}}^{\vee}(w_{ij}, x_j^{\omega}) \quad (13)$$

Teorema 1 (Teorema 4.20 en [24]). Sea $\{(\mathbf{x}^{\mu}, \mathbf{y}^{\mu}) \mid \mu = 1, 2, \dots, p\}$ el conjunto fundamental de una memoria heteroasociativa alfa-beta

representada por \mathbf{W} . Si ω es un valor de índice arbitrario tal que $\omega \in \{1, 2, \dots, p\}$, y si además para cada $i \in \{1, 2, \dots, m\}$ se cumple que $\exists j = j_0 \in \{1, 2, \dots, n\}$, el cual depende de ω y de i tal que $w_{j_0} = \alpha_{\mathbb{R}}(y_i^\omega, x_{j_0}^\omega)$, entonces la recuperación $\mathbf{W}\nabla_{\beta_{\mathbb{R}}} \mathbf{x}^\omega$ es correcta en términos de que $\mathbf{W}\nabla_{\beta_{\mathbb{R}}} \mathbf{x}^\omega = \mathbf{y}^\omega$.

Demostración. Los detalles de esta demostración pueden ser consultados en Yáñez [24].

2.1.2 Memorias heteroasociativas Alfa-Beta máx

El algoritmo de las memorias heteroasociativas Alfa-Beta tipo máx, se realiza de la misma manera que el de las mín que se muestra en la sección 2.1.1. La diferencia consiste en intercambiar todos los operadores mínimo \wedge por máximo \vee y todos los operadores máximo \vee por mínimo \wedge . La memoria generada será representada con una \mathbf{M} en lugar de la \mathbf{W} con la que se representan las tipo mín.

3 Transformada Alfa-Beta

En un sistema de compresión de imágenes, la fase de transformación tiene la función de facilitar la compresión de información en las etapas subsecuentes.

La transformada Alfa-Beta presentada en este artículo se aplica sobre sub-bloques de una imagen generando una MAABI heteroasociativa en cada uno de estos sub-bloques, a partir de una matriz de transformación dada.

3.1 Definiciones, lemas y teoremas

A continuación daremos las definiciones necesarias para explicar el algoritmo.

Definición 1. Sea $\mathbf{A} = [a_{ij}]$ una matriz de dimensión $m \times n$ que representa una imagen, y sea $\mathbf{sb} = [sb_{ij}]$ una matriz de $d \times d$ que

representa un **sub-bloque de la imagen A**, tal que

$$sb_{ij} = a_{rtj} \tag{14}$$

donde $i, j = 1, 2, \dots, d$, $r = 1, 2, \dots, m$, $t = 1, 2, \dots, n$ y $sb_{ij} = a_{rtj}$ representa el valor del píxel dado por las coordenadas $(r+i, t+j)$ donde (r, t) y $(r+d, t+d)$ son el principio y el final del sub-bloque, respectivamente.

Definición 2. Sea $\mathbf{sb} = [sb_{ij}]$ un sub-bloque de imagen y sea $\mathbf{vi} = [vi_i]$ un vector de dimensión d . La i -ésima fila del sub-bloque \mathbf{sb} es un vector de **imagen vi** tal que

$$\mathbf{vi} = [sb_{i1}, sb_{i2}, \dots, sb_{id}] \tag{15}$$

donde $i = 1, 2, \dots, d$.

Definición 3. Sea $\mathbf{A} = [a_{ij}]$ una matriz de dimensión $m \times n$ que representa una imagen. El valor designado por ε representa un valor mayor al máximo que puede tomar un píxel de la imagen, como sigue:

$$\varepsilon > \bigvee_{\forall i, j} a_{ij} \tag{16}$$

Definición 4. Sea $\mathbf{vt} = [vt_i]$ un vector de dimensión d llamado **vector de transformación**. Al tener dos o más vectores de transformación, se generan de la siguiente manera:

$$vt_j^i = \begin{cases} 0 & i \neq j \\ \varepsilon & i = j \end{cases} \tag{17}$$

Definición 5. Sea $\mathbf{vt} = [vt_i]$ un vector de transformación de dimensión d . El conjunto formado por los d vectores de transformación

$\{vt_1, vt_2, \dots, vt_d\}$ conforma la llamada **matriz de transformación** $\mathbf{mt} = [mt_{ij}]_{d \times d}$ donde la i -ésima fila de la matriz \mathbf{mt} queda representada por el i -ésimo vector de transformación vt_i . Por lo tanto, la ij -ésima componente de la matriz de transformación \mathbf{mt} está dada por:

$$mt_{ij} = vt_j^i \mid i, j = 1, 2, \dots, d \quad (18)$$

Definición 6. Sea $\mathbf{A} = [a_{ij}]$ una matriz de dimensión $m \times n$ que representa una imagen, sea $\mathbf{sb} = [sb_{ij}]$ un sub-bloque de la imagen \mathbf{A} , obteniendo como resultado κ sub-bloques de imagen $\mathbf{sb}^\mu \mid \mu = 1, 2, \dots, \kappa$ donde $\kappa = (m/d) \cdot (n/d)$ y sea $\mathbf{mt} = [mt_{ij}]$ una matriz de transformación de dimensión $d \times d$. Una transformada alfa-beta (**TAB**)

$$\begin{aligned} \mathbf{MAABI}_{\min}^{xy} &= \bigwedge_{\mu=1}^d \left[\mathbf{vi}^{\omega_\mu} \otimes_{\mathbb{R}} (\mathbf{vt}^\mu)^t \right] = \\ &= [w_{ij}^{xy}]_{d \times d} = \bigwedge_{\mu=1}^d \left[vl_i^{\omega_\mu} - vt_j^\omega + 1 \right] \end{aligned} \quad (21)$$

donde $\omega = 1, 2, \dots, \kappa$, $i, j = 1, 2, \dots, d$.

Definición 8. Sea **TAB** una transformada alfa-beta de una imagen, $\mathbf{mt} = [mt_{ij}]$ una matriz de transformación la **TAB** está conformada por κ **MAABI** distribuidas matricialmente como:

$$\mathbf{TAB}_\tau(\mathbf{A}, \mathbf{mt}) \rightarrow O\{\mathbf{MAABI}_\tau^{xy}\} \quad (22)$$

donde $x = 1, 2, \dots, \lambda$, $y = 1, 2, \dots, \eta$, con $\lambda = m/d$, $\eta = n/d$, $\kappa = \lambda \cdot \eta$ y τ indica si las **MAABI** son del tipo *máx* o *mín*. Se define como **transformada alfa-beta inversa (TABI)** al proceso que aplica la fase de recuperación de una **MAABI** entre los vectores de transformación \mathbf{vt}^μ , considerados como patrones de entrada, y las **MAABI** que conforman la **TAB**, como sigue:

$$\begin{aligned} \mathbf{TABI}(\mathbf{MAABI}^{ij}, \mathbf{mt}) &\rightarrow O\{\mathbf{sb}^{ij}\} \\ O\{\mathbf{sb}^{ij}\} &= \begin{pmatrix} \mathbf{sb}^{11} & \mathbf{sb}^{12} & \dots & \mathbf{sb}^{1\eta} \\ \mathbf{sb}^{21} & \mathbf{sb}^{22} & \dots & \mathbf{sb}^{2\eta} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{sb}^{\lambda 1} & \mathbf{sb}^{\lambda 2} & \dots & \mathbf{sb}^{\lambda \eta} \end{pmatrix} \end{aligned} \quad (23)$$

donde $i = 1, 2, \dots, \lambda$, $j = 1, 2, \dots, \eta$, $\lambda = m/d$ y $\eta = n/d$. En la cual el operador $O\{\cdot\}$ significa que las matrices \mathbf{sb}^{ij} guardan las mismas posiciones en la **TABI** que las **MAABI** en la **TAB**. Sabemos que realizar la **TABI** es posible gracias a que la imagen que transformamos es un conjunto de **MAABI** y que además dado que la \mathbf{mt} es de antemano conocida, siempre está disponible para el proceso inverso de la transformación.

Dado que existe una **TAB**_{min} es necesario definir una **TABI**_{min} como sigue.

Definición 9. Sea **TABI** una transformada alfa-beta inversa. La **transformada alfa-beta mín (TAB**_{min}**)** está definida por:

$$\mathbf{TABI}_{\min}(\mathbf{MAABI}_{\min}^{xy}, \mathbf{mt}) \rightarrow O\{\mathbf{sb}_{\min}^{xy}\} \quad (24)$$

$$\mathbf{sb}_{\min}^{xy} = \mathbf{vi}^{(xy)\mu} \mid \mu = 1, 2, \dots, d$$

$$\mathbf{vi}^{(xy)\mu} = \mathbf{MAABI}_{\min}^{xy} \nabla_{\beta_x} \mathbf{vi}^\mu = [vl_i^{(xy)\mu}]_d \quad (25)$$

$$vl_i^{(xy)\mu} = \bigvee_{j=1}^d \beta_{\mathbb{R}}^\vee(w_{ij}^{xy}, vt_j^\mu) \quad (26)$$

donde xy refiere a cuál de los κ sub-bloques de imagen \mathbf{sb} pertenecen los vectores de imagen \mathbf{sb} , es decir $\mathbf{vi}^{(xy)\mu}$ representa el μ -ésimo patrón del sub-bloque de imagen con índice xy .

Lema 1. Sean \mathbf{vt}^μ los patrones de entrada y \mathbf{vi}^μ sus correspondientes patrones de salida con $\mu = 1, 2, \dots, d$, y sea \mathbf{W} una **MAABI**_{min}. Al calcular todas las matrices intermedias \mathbf{w}^μ que resultan

de cada asociación $(\mathbf{x}^\mu, \mathbf{y}^\mu)$, podemos decir que la μ -ésima columna de la matriz \mathbf{w}^μ es menor por un valor de ε , que las $d-1$ columnas restantes, así pues:

$$w_{i\mu}^\mu = w_{ik}^\mu - \varepsilon, k \neq \mu \quad (27)$$

Demostración. Sabemos que la dimensión de los patrones \mathbf{vt}^μ determina el número de columnas de la memoria asociativa, y la dimensión de los patrones \mathbf{vi}^μ sus filas. En la fase de aprendizaje, la memoria asociativa se construye como sigue:

$$\mathbf{W} = \bigwedge_{j=1}^d \left[\mathbf{vi}^\mu \otimes_{\mathbb{R}} (\mathbf{vt}^\mu)^t \right]_{d \times d} \quad (28)$$

Esto quiere decir que se calculan μ matrices intermedias de tal manera que

$$\begin{aligned} \mathbf{w}^\mu &= \left[\mathbf{vi}^\mu \otimes_{\mathbb{R}} (\mathbf{vt}^\mu)^t \right] \\ w_{ij}^\mu &= \left[vi_i^\mu - vt_j^\mu + 1 \right] \end{aligned} \quad (29)$$

Si elegimos arbitrariamente una matriz intermedia \mathbf{w}^ω , $\omega \in \{1, 2, \dots, d\}$, tomando en cuenta la *Definición 3* y la *Definición 4*, entonces $vt_{j=\omega}^\omega = \varepsilon$ y los valores de las componentes $vt_{j \neq \omega}^\omega = 0$. Teniendo esto, obtenemos la expresión

$$w_{i\omega}^\omega = y_i^\omega - x_{j=\omega}^\omega + 1 = y_i^\omega - \varepsilon + 1 \quad (30)$$

para la ω -ésima columna de la ω -ésima matriz intermedia, y la expresión

$$w_{ik}^\omega = y_i^\omega - x_k^\omega + 1 = y_i^\omega - 0 + 1 = y_i^\omega + 1, k \neq \omega \quad (31)$$

para todos los demás índices de columna de dicha matriz intermedia. Esto indica que para la

pareja (x^ω, y^ω) , la ω -ésima columna de la matriz $\mathbf{w}^\omega = \mathbf{y}^\omega \otimes_{\mathbb{R}} (\mathbf{x}^\omega)^t$ será menor que las otras columnas por un valor de ε , sustituyendo la expresión (31) en (30):

$$w_{i\omega}^\omega = w_{ik}^\omega - \varepsilon, k \neq \omega \quad (32)$$

Como la matriz intermedia \mathbf{w}^ω , $\omega \in \{1, 2, \dots, d\}$ se eligió de manera arbitraria, se garantiza que aplica para todas las asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$, $\mu = 1, 2, \dots, d$, de tal forma:

$$w_{i\mu}^\mu = w_{ik}^\mu - \varepsilon, k \neq \mu \quad (33)$$

Teorema 2. Sean \mathbf{vt}^μ los patrones de entrada y \mathbf{vi}^μ sus correspondientes patrones de salida con $\mu = 1, 2, \dots, d$, y sea \mathbf{W} una **MAABI**_{min}. Al realizar la fase de aprendizaje, el patrón \mathbf{vi}^μ se codificará completamente en la μ -ésima columna de la memoria \mathbf{W} . De esta forma basta la siguiente expresión para generar dicha memoria:

$$w_{i\mu} = vi_i^\mu - vt_\mu^\mu + 1, \mu, i = 1, 2, \dots, d \quad (34)$$

Demostración. Sabemos que la dimensión de los patrones \mathbf{vt}^μ determina el número de columnas de la memoria asociativa, y la dimensión de los patrones \mathbf{vi}^μ sus filas. De acuerdo con el *Lema 1*, la μ -ésima columna de cada matriz $\mathbf{w}^\mu = \mathbf{vi}^\mu \otimes_{\mathbb{R}} (\mathbf{vt}^\mu)^t$ es menor por un valor de ε , que las $d-1$ columnas restantes: $w_{i\mu}^\mu = w_{ik}^\mu - \varepsilon, k \neq \mu$. En la fase de aprendizaje, la ij -ésima componente de \mathbf{W} se obtiene mediante:

$$w_{ij} = \bigwedge_{\mu=1}^d \left[vi_i^\mu - vt_j^\mu + 1 \right]_{d \times d} \quad (35)$$

Entonces debemos demostrar que la μ -ésima columna de la μ -ésima matriz intermedia es menor que la μ -ésima columna de las $d-1$ matrices intermedias restantes, es decir que

$$w_{i\mu}^\mu < w_{i\mu}^{k \neq \mu} \quad (36)$$

Para ello, elegimos arbitrariamente una matriz intermedia w^ω , $\omega \in \{1, 2, \dots, d\}$, y los valores de su ω -ésima columna deben ser mayores que los valores de la ω -ésima columna de las matrices intermedias w^θ , $\theta \neq \omega$, es decir que

$$w_{i\omega}^\omega < w_{i\omega}^\theta \quad (37)$$

Tomando en consideración que $w_{i\omega}^\omega = y_i^\omega - x_\omega^\omega + 1$ y $w_{i\omega}^\theta = y_i^\theta - x_\omega^\theta + 1$, sustituyendo en la expresión (37), tenemos:

$$v_i^\omega - v_\omega^\omega + 1 < v_i^\theta - v_\omega^\theta + 1 \quad (38)$$

La *Definición 4* dice que $v_\omega^\omega = \varepsilon$ y $v_\omega^\theta = 0$, que sustituyendo en (38) es $v_i^\omega - \varepsilon + 1 < v_i^\theta - 0 + 1$, es decir

$$v_i^\omega - \varepsilon < v_i^\theta \quad (39)$$

pero por la *Definición 3* sabemos que $\varepsilon > \bigvee_{\forall \mu, i} v_i^\mu$,

por ejemplo $\varepsilon = \bigvee_{\forall \mu, i} v_i^\mu + 1$, y sustituyendo en (39):

$$v_i^\omega - \left(\bigvee_{\forall \mu, i} v_i^\mu + 1 \right) < v_i^\theta \quad (40)$$

lo cual demuestra la veracidad de $v_i^\omega - v_\omega^\omega + 1 < v_i^\theta - v_\omega^\theta + 1$ y por lo tanto de $w_{i\omega}^\omega < w_{i\omega}^\theta$. Esto indica que la matriz w^ω aporta su ω -ésima

columna completa a la memoria asociativa \mathbf{W} tal que

$$w_{i\omega} = v_i^\omega - v_\omega^\omega + 1 \quad (41)$$

Como ω se eligió arbitrariamente, podemos concluir que aplica para todas las matrices intermedias, así pues

$$w_{i\mu} = v_i^\mu - v_\mu^\mu + 1 \quad (42)$$

3.2 Algoritmo de la transformada Alfa-Beta

El algoritmo de la transformada alfa-beta está descrito en los siguientes puntos:

1. Al igual que los métodos de transformación tradicionales, la transformada alfa-beta se aplica individualmente sobre bloques de dimensión $d \times d$ de una imagen. Dividimos la imagen descrita por la matriz $\mathbf{A} = [a_{ij}]$ de dimensión $m \times n$ en $\kappa = (m/d) \times (n/d)$ sub-bloques de imagen $\mathbf{sb}^\omega \mid \omega = 1, 2, \dots, \kappa$ donde m y n representan el alto y ancho de la imagen, respectivamente; a_{ij} representa el valor del ij -ésimo píxel con $a \in \{1, 2, \dots, 2^L - 1\}$ siendo L el número de bits necesarios para representar el valor de un píxel.
2. Inicializamos la **TAB** colocando todos sus valores en 0.
3. Creamos la matriz de transformación $\mathbf{mt} = [mt_{ij}]_{d \times d}$ a utilizar en la **TAB** de la forma indicada en la *Definición 5*.
4. Aplicamos las memorias asociativas alfa-beta para imágenes sobre cada sub-bloque de imagen de acuerdo con la *Definición 6* y la *Definición 7*, adecuándola con el *Teorema 2*:

$$\mathbf{MAABI}_{m\omega}^{xy} = v_i^\mu - v_\mu^\mu + 1 = [w_{ij}]_{d \times d}^{xy} \mid \omega = 1, 2, \dots, \kappa \quad (43)$$

5. Como resultado de los pasos anteriores, obtendremos κ **MAABI** de dimensión $d \times d$,

ordenadas por el operador $O\{\cdot\}$, las cuales en conjunto conforman la imagen transformada por la correspondiente **TAB**, de tal forma que

$$\mathbf{TAB}_{\min}(\mathbf{A}, \mathbf{mt}) \rightarrow O\{\mathbf{MAABI}_{\min}^{xy}\} \quad (44)$$

donde $x=1,2,\dots,\lambda$, $y=1,2,\dots,\eta$, $\lambda=m/d$, $\eta=n/d$ y $\kappa=\lambda \times \eta$.

3.3 Algoritmo de la transformada Alfa-Beta inversa

El proceso de una transformada alfa-beta inversa se describe a continuación:

1. La **TAB** está formada por κ **MAABI**_{min} organizadas de forma matricial. Tomamos por separado las κ **MAABI**_{min} con $\kappa=(m/d) \times (n/d)$ donde m y n representan el alto y ancho de la imagen transformada, respectivamente.
2. Aplicamos el proceso de la **TABI** tomando cada una de las **MAABI**_{min} y la matriz de transformación **mt**, obteniendo d vectores de imagen $\mathbf{vi}^\mu = [vi_i^\mu]_d \mid \mu=1,2,\dots,d$.

$$vi_i^{(xy)\mu} = \bigvee_{j=1}^d (vi_j^\mu - |w_{ij}^{xy}| - 1) \quad (45)$$

Al agrupar estos vectores de imagen, se forman los sub-bloques de imagen $\mathbf{sb}^{xy} = vi_i^{(xy)\mu} \mid \mu=1,2,\dots,d$ donde xy determina cuál de los κ sub-bloques de imagen se ha recuperado.

3. Por último, agrupamos los κ sub-bloques de imagen mediante el operador $O=\{\cdot\}$ obteniendo así la imagen recuperada **A**.

$$\mathbf{A} = O\{\mathbf{sb}^{xy} \mid x=1,2,\dots,\lambda, y=1,2,\dots,\eta\} \quad (46)$$

Debido a que cada vez que formamos la matriz de transformación utilizamos los máximos del sub-bloque en cuestión, es necesario guardar un vector que contenga dichos máximos. Para realizar una medida de efectividad de la transformada propuesta, vamos a experimentar realizando las transformadas y comparando la entropía de la imagen original, con la entropía de las imágenes transformadas. En la teoría de la información de Shannon [Shannon, 1948], la medida de la información se define como un proceso probabilístico, considerando la imagen como una fuente de información, su información promedio o entropía se define como

$$H(S) = -\sum_i P(s_i) \log_2(P(s_i)) \quad (47)$$

4 Resultados

En esta sección se presentan los resultados obtenidos al aplicar la TAB a un conjunto variado de imágenes de prueba bastante conocidas y utilizadas en el ámbito del procesamiento de imágenes.

Dichas imágenes son en escala de grises, de dimensiones 512x512 píxeles de 8 bits/píxel. Se experimentó con sub-bloques de dimensión $d=4$, $d=8$ y $d=16$. Para comparar la imagen transformada contra la original, se utiliza la entropía de Shannon como medida para conocer la cantidad de bits contenidos en cada imagen, y se muestra el tiempo de ejecución del algoritmo.

La TAB se implementó en Matlab 7.4 (2007 2a) sobre una computadora personal, con un procesador Intel Pentium D a 3.40 GHz, 2048 MB de memoria RAM y 109.5 GB de espacio libre en disco duro. Se usó el sistema operativo Windows XP, de Microsoft.

Algunos ejemplos de las imágenes transformadas se pueden encontrar de la Fig. 3 a la Fig. 10. En la información de la Tabla 1 a la Tabla 3, pueden verse los datos de los experimentos.

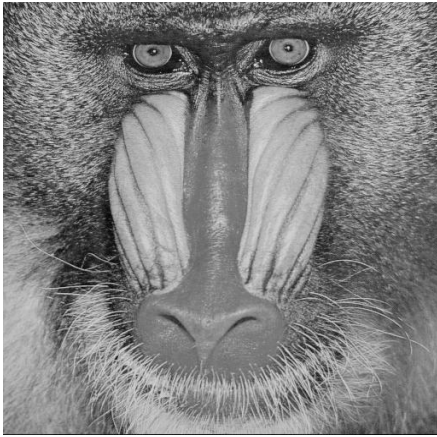


Fig. 3. Imagen "Baboon"

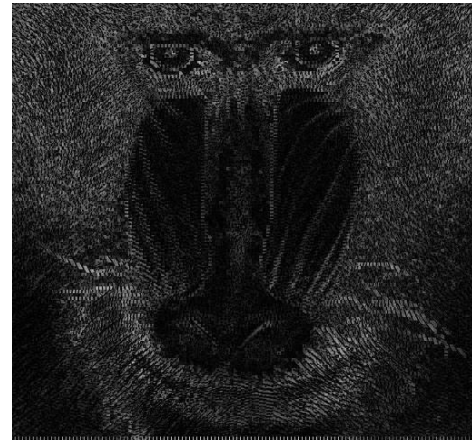


Fig. 4. Imagen "Baboon" transformada



Fig. 5. Imagen "Goldhill"



Fig. 6. Imagen "Goldhill" transformada utilizando sub-bloques de 8x8



Fig. 7. Imagen "Tiffany"

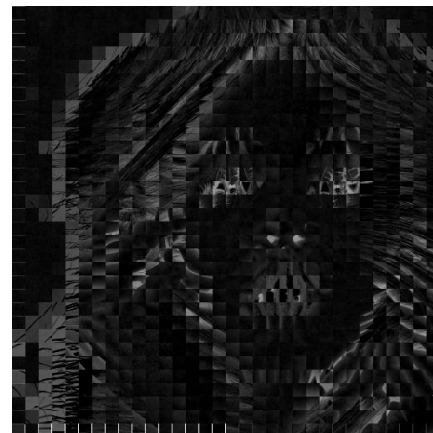


Fig. 8. Imagen "Tiffany" transformada utilizando sub-bloques de 16x16



Fig. 9. Imagen "Lena"

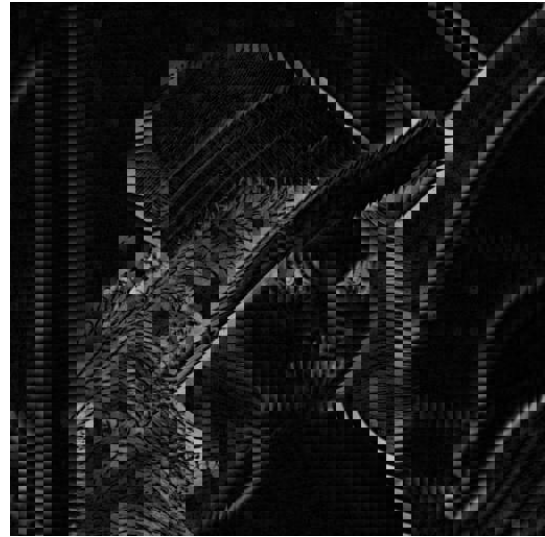


Fig. 10. Imagen "Lena" transformada utilizando sub-bloques de 8x8

Estos datos son la entropía de la imagen original (EnO), la entropía de la imagen transformada (EnT), el tiempo de procesamiento en segundos de la transformación de la imagen (TAB t) y el tiempo de procesamiento de la transformada inversa (TABI t).

En la Tabla 1 pueden encontrarse los

resultados de la transformación utilizando sub-bloques de tamaño 4x4, la Tabla 2 muestra los resultados al realizar la transformación utilizando sub-bloques de tamaño 8x8, en la Tabla 3 se observan los resultados para imágenes tomando sub-bloques de tamaño 16x16.

Tabla 1. Resultados utilizando sub-bloques de 4x4

Imagen 512x512	EnO	EnT	TAB t	TABI t
Baboon	7.3579	6.4604	1.20 s	0.84 s
Barbara	7.4664	5.6765	1.15 s	0.84 s
Boat	7.1914	5.5520	1.21 s	0.88 s
Couple	7.2010	5.3092	1.20 s	0.86 s
Elaine	7.5060	5.1970	1.19 s	0.92 s
F-16	6.7166	4.8885	1.19 s	0.91 s
Goldhill	7.4778	5.4947	1.13 s	0.88 s
Lena	7.4456	5.1036	1.15 s	0.88 s
Peppers	7.5715	5.1306	1.18 s	0.87 s
Tiffany	6.6474	4.8689	1.15 s	0.87 s

Tabla 2. Resultados utilizando sub-bloques de 8x8

Imagen 512x512	EnO	EnT	TAB t	TABI t
Baboon	7.3579	6.9273	0.33 s	0.11 s
Barbara	7.4664	6.3335	0.34 s	0.07 s
Boat	7.1914	6.2688	0.35 s	0.09 s
Couple	7.2010	6.0276	0.34 s	0.07 s
Elaine	7.5060	5.8841	0.32 s	0.10 s
F-16	6.7166	5.5727	0.53 s	0.08 s
Goldhill	7.4778	6.2366	0.37 s	0.05 s
Lena	7.4456	5.8906	0.36 s	0.09 s
Peppers	7.5715	5.9642	0.34 s	0.12 s
Tiffany	6.6474	5.5107	0.32 s	0.07 s

Tabla 3. Resultados utilizando sub-bloques de 16x16

Imagen 512x512	EnO	EnT	TAB t	TABI t
Baboon	7.3579	7.1490	0.13 s	0.21 s
Barbara	7.4664	6.8774	0.13 s	0.20 s
Boat	7.1914	6.8104	0.14 s	0.18 s
Couple	7.2010	6.5776	0.13 s	0.21 s
Elaine	7.5060	6.5101	0.13 s	0.20 s
F-16	6.7166	6.0468	0.13 s	0.25 s
Goldhill	7.4778	6.7533	0.09 s	0.18 s
Lena	7.4456	6.5461	0.16 s	0.20 s
Peppers	7.5715	6.7611	0.15 s	0.17 s
Tiffany	6.6474	6.0072	0.14 s	0.22 s

Tabla 4. Comparación entre TAB, TM, DCT y DWT

	TAB	TM	DCT	DWT
Precisión	Enteros	Enteros	Punto flotante	Punto flotante
Tipo de operaciones	+,-	-,^	+,x,/ e,cos	+,x, /,√
Menor entropía	Sí	No	No	No

Tabla 5. Comparación entre los procesos inversos de la TAB, TM, DCT y DWT

	TAB	TM	DCT	DWT
Precisión	Enteros	Enteros	Punto flotante	Punto flotante
Tipo de operaciones	-,√	+,√	+,x,/ e,cos	+,x, /,√
Reversible	Sí	Sí	Sí	Sí

5 Conclusiones

Se presenta la teoría de las MAABI que se basan en la modificación de los operadores Alfa y Beta originales para poder trabajar en el campo de los números reales. Estas memorias resultan en la creación de la TAB, cuya teoría y sustento matemático consta de 9 definiciones, así como la demostración de 1 lema y 2 teoremas que se presentan en este artículo. La TAB es reversible, por lo que al realizar la TABI, que es el proceso de transformación inversa del método propuesto, obtendremos la imagen original sin alteraciones. Esto está demostrado en el Teorema 1. Adicionalmente se realizó una modificación en el funcionamiento de las MAABI cuando son utilizadas bajo las condiciones que requiere una TAB, de tal manera que se simplifica su fase de aprendizaje. Dicha simplificación consiste en calcular una sola matriz que genere la memoria como está demostrado en el Teorema 2, en lugar de calcular tantas matrices como patrones de entrada existan y además aplicar el operador mínimo elemento a elemento entre cada una de estas matrices, como se indica en la expresión (21). Por lo que el ahorro sustancial en el número de operaciones requeridas por el algoritmo constituye una de las ventajas sobre la TM.

Otra de las ventajas de la transformada presentada en este artículo es la sencillez en sus operaciones en comparación con la DCT y la DWT, además de que estas últimas requieren una precisión de punto flotante y la TAB trabaja con enteros al igual que la TM. La TM al igual que la DCT y la DWT no ofrecen compresión alguna, siendo que la TAB ofrece una disminución en la medida de entropía entre la imagen transformada y la imagen original. Por lo que se infiere que al realizar las siguientes etapas del proceso de compresión de imágenes, utilizando la TAB, se obtendrán resultados competitivos y mejorías en la compresión de las imágenes. Todos estos datos podemos verlos resumidos en la Tabla 4 y la Tabla 5.

El uso de la TAB representa un bajo costo computacional con algoritmos de alta velocidad de procesamiento (y aún puede mejorarse al implementarse en otro tipo de lenguajes, como C++). Además de que, como se mostró en los resultados, al transformar la imagen obtenemos una medida de la entropía menor a la original, lo que es una característica deseable para el proceso de transformación de datos en un sistema de compresión de imágenes. Aun así, se requiere trabajar en la inclusión y pruebas de los procesos subsecuentes en la compresión de

imágenes para poder realizar una comparación directa en la relación de compresión contra los demás modelos aquí mencionados; aspecto que se tomará en cuenta para trabajos futuros en el área.

Agradecimientos

Los autores agradecen al Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, y al CIC), al CONACyT, al SNI, y al ICyTDF (apoyos PIUTE10-77 y PICSO10-85) por su apoyo económico para el desarrollo de este trabajo.

Referencias

1. **Ahmed, N., Natarajan, T., & Rao, K.R. (1974).** Discrete cosine transform. *IEEE Transactions on Computers*, 23(1), 90–93.
2. **Antonini, M., Barlaud, M., Mathieu, P., & Daubechies, I. (1992).** Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2), 205–220.
3. **Arai, Y., Agui, T., & Nakajima, M. (1988).** A fast DCT-SQ scheme for image. *IEICE Transactions*, E71(11), 1095–1097.
4. **Chen, W.H., Smith, C., & Fralick, S. (1977).** A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on communications*, 25(9), 1004–1009.
5. **Daubechies, I. (1990).** The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on information theory*, 36(5), 961–1005.
6. **Guzmán, E., Pogrebnyak, O., Yáñez, C., & Sánchez, L.P. (2008).** Morphological transform for image compression. *EURASIP Journal on advances in signal processing*, 2008, Article ID 426580.
7. **Guzmán, E. (2008).** *Compresión de imágenes mediante memorias asociativas*. Tesis Doctoral. Instituto Politécnico Nacional, Centro de Investigación en Computación, México, D.F.
8. **Haralick, R.M. (1976).** A storage efficient way to implement the discrete cosine transform. *IEEE Transactions on computers*, C-25(7), 764–765.
9. **Hassoun, M.H. (1993).** *Associative Neural Memories: theory and implementation*. New York: Oxford University Press.
10. **Islam, A. & Pearlman, W. A. (1999).** An embedded and efficient low-complexity hierarchical image coder. *Visual communications and image processing'99 (SPIE 3653)*, San Jose, CA, 294–305.
11. **JPEG-2000 Final committee draft version 1.0**, ISO/IEC JTC1/SC 29/WG1 N1646R, 2000.
12. **Generic Coding of Moving Pictures and Associated Audio**, ISO/IEC 13818-2, ITU-T, 1995.
13. **Rao, K. R. & Yip, P. (1990).** *Discrete cosine transform: algorithms, advantages, applications*. Boston: Academic Press.
14. **Ritter, G.X., Sussner, P., & Díaz-de-León, J.L. (1998).** Morphological Associative Memories. *IEEE Transactions on neural networks*, 9(2), 281–293.
15. **Said, A. & Pearlman, W.A. (1996).** A new, fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on circuits and systems for video technology*, 6(3), 243–250.
16. **Shannon, C.E. (1948).** A mathematical theory of communication. *Bell system technical journal*, 27, 379–423, 623–656.
17. **Shapiro, J. M. (1993).** Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on signal processing*, 41(12), 3445–3462.
18. **Sherlock, B.G. & Monro, D.M. (1995).** Algorithm 749: Fast Discrete Cosine Transform. *ACM Transactions on mathematical software*, 21(4), 372–378.
19. **Taubman, D. (2000).** High performance scalable image compression with EBCOT. *IEEE Transactions on image processing*, 9(7), 1158–1170.
20. **Tseng, B.D. & Miller, W.C. (1978).** On computing the discrete cosine transform. *IEEE Transactions on computers*, 27(10), 966–968.
21. **Wallace, G. K. (1991).** The JPEG still-picture compression standard. *Communications of the ACM*, 34(4), 30–44.
22. **Xiong, Z., Ramchandran, K., Orchard, M.T., & Zhang Y.-Q. (1999).** A comparative study of DCT- and wavelet-based image coding. *IEEE Transactions on circuits and systems for video technology*, 9(5), 692–695.
23. **Yáñez, C. & Díaz-de-León, J.L. (2003).** *Introducción a las Memorias Asociativas*. México: Instituto Politécnico Nacional, Centro de Investigación en Computación.
24. **Yáñez, C. (2002).** *Memorias asociativas basadas en relaciones de orden y operadores binarios*. Tesis doctoral, Instituto Politécnico Nacional, Centro de Investigación en Computación, México, D.F.



Antonio Alarcón Paredes

recibió el grado de Maestro en Ciencias en Ingeniería de Cómputo, en 2009, por el Centro de Investigación en Computación del Instituto Politécnico Nacional; donde actualmente se encuentra realizando estudios de Doctorado en Ciencias de la Computación. Sus áreas de interés son: memorias asociativas, reconocimiento de patrones y procesamiento/compresión de imágenes.



Oleksiy Pogrebnyak

es Doctor en Ciencias en Sistemas Radiotécnicos, por el Instituto de Aviación de Kharkov (ahora Universidad Nacional Aeroespacial), Ucrania, en 1991. Actualmente, trabaja en el Centro de Investigación en Computación del Instituto Politécnico Nacional, Mexico. Sus áreas de interés incluyen el filtrado y compresión de imágenes/señales digitales y percepción remota.



Amadeo José Argüelles Cruz

es Doctor en Ciencias de la Computación por el Centro de Investigación en Computación, del Instituto Politécnico Nacional, en 2007. Sus áreas de interés son redes neuronales sin pesos, memorias asociativas, reconocimiento de patrones.

Article received on 25/04/2011, accepted on 20/12/2011.