

Obtención de Rasgos Preponderantes en Imágenes de Escena Natural mediante un Modelo de Atención Selectiva

Visual Saliency Detection in Natural Scene Images using a Selective Attention Model

Yesenia González¹, Alan Solano², Herón Molina³, Sergio Garduza⁴, Benito Granados⁵

Recibido: mayo 2017

Aprobado: agosto 2017

Resumen.- En este artículo se presenta la extracción de rasgos preponderantes en imágenes de escena natural. Las imágenes son procesadas utilizando como base los modelos de color RGB, HSI y CMY, para posteriormente utilizar combinaciones de las componentes de color como características de entrada a un modelo de atención selectiva basado en la aplicación de un filtro especializado de Gabor en 2 dimensiones, con el cual se obtienen algunos de los rasgos (bordes y contrastes sobresalientes), que después son resaltados por una etapa de agrupamiento de datos y una etapa de red neuronal artificial de competencia. Los resultados de las simulaciones muestran que el sistema es capaz de encontrar los rasgos preponderantes en escenas sencillas y muestra resultados alentadores en escenas complejas. Para las pruebas se utilizaron imágenes en formato de color RGB de 640×480 píxeles (VGA). La implementación se realizó en el lenguaje MATLAB®.

Palabras clave: Obtención de rasgos preponderantes; atención selectiva; filtro de Gabor en 2D; agrupamiento de datos; red neuronal de competencia.

Summary.- This article presents visual saliency detection in natural scene images. Images are processed using RGB, HSI and CMY color models and use some combinations of color components to feed a selective attention model based on the application of a two-dimensional specialized Gabor filter, which gives some of the features (like edges and outstanding contrasts), to be later highlighted by a clustering stage and a competitive artificial neural network stage. The simulations results show that the system is able to perform visual saliency detection in simple scenes and show encouraging results in complex scenes. For the tests were used images in RGB color format of 640×480 pixels (VGA). The implementation was made in the MATLAB® language.

Keywords: Visual saliency detection; selective attention; 2D Gabor filter; data clustering; competitive neural network.

1. Introducción.- La atención selectiva en los humanos se ha definido como el mecanismo en el cerebro que determina qué parte de la multitud de datos sensoriales que se están adquiriendo en determinado instante es de mayor interés [1]. Debido a que la cantidad de datos que se reciben es muy alta para realizar un procesamiento a detalle de toda la información y además generar acciones como respuesta, el cerebro prioriza a partir de ese mecanismo.

¹ Doctora en Ciencias. UPIITA - Instituto Politécnico Nacional. ygonzalez@ipn.mx (Autor de contacto)

² Ingeniero. UPIITA - Instituto Politécnico Nacional. sco1192@gmail.com

³ Doctor en Ciencias. CIC- Instituto Politécnico Nacional. hmolina@cic.ipn.mx

⁴ Maestro en Ciencias. UPIITA-Instituto Politécnico Nacional. sgarduzag@ipn.mx

⁵ Maestro en Ciencias. CINVESTAV - Instituto Politécnico Nacional. bgranadosr@cinvestav.mx

Con respecto a la atención visual en humanos, el orden en que una escena es analizada está determinada por el mecanismo de atención selectiva. La atención visual es un proceso que se cree ocurre antes del reconocimiento de los objetos en una escena [2] y puede incluir un mecanismo de selección y la noción de relevancia [3]. La atención visual puede categorizarse en dos principales modelos [1, 3]: ascendente y descendente. El modelo ascendente únicamente se basa en información contenida en la escena visual, mientras que el modelo descendente considera factores cognitivos tales como conocimiento, expectativa, recompensa y objetivos.

En este trabajo se aborda un modelo de atención visual ascendente para la extracción de rasgos preponderantes basado en la teoría de la integración de rasgos (FIT, por sus siglas en Inglés) [3, 4], donde la imagen de entrada es procesada utilizando la combinación de distintos modelos de color [5] y es descompuesta en mapas de rasgos a través de la implementación de un filtro de Gabor en 2 dimensiones (2D) [6]. En cada uno de los mapas de rasgos distintas localidades espaciales compiten entre sí para obtener mayor preponderancia, de tal manera que únicamente las regiones que se destacan localmente de su entorno persisten. La salida del sistema es un único mapa de preponderancias resultante de otros procesos de competencia (que se explicarán más adelante) entre los distintos mapas de rasgos. Los rasgos preponderantes de una escena pueden referirse a objetos o regiones que sobresalen o destacan en relación a su vecindad [3]. Este artículo se ha organizado de la siguiente manera: en la sección 2 se realiza la descripción general del sistema para después describir a detalle cada una de las etapas que lo conforman. En la sección 3 se presentan los valores de parámetros utilizados y los resultados obtenidos. En la sección 4 se presentan las conclusiones.

2. Descripción del sistema.- Aunque el sistema propuesto toma como base ciertas etapas utilizadas en arquitecturas previamente reportadas [2, 5] con respecto al procesamiento de las imágenes haciendo combinaciones de diferentes componentes de color y a la utilización del filtro de Gabor en 2D, en este trabajo se han introducido nuevas etapas de extracción de características que utilizan los métodos de agrupamiento de datos [7] en combinación con redes neuronales artificiales (RNA) de competencia [7, 8]. El sistema propuesto consiste en ingresar una imagen en formato de color RGB a una etapa de transformación de color, que combina matrices de datos a partir de los modelos RGB, HSI e incluso la componente en amarillo (Y) del modelo de color CMY [9], para crear mapas de color [10]. A continuación, los mapas de color obtenidos pasan a través de un filtro de Gabor en 2 dimensiones, que al variar algunos de sus parámetros genera una serie de mapas de rasgos (MR) o mapas de características. Si se considera utilizar M mapas de color y se realizan N variaciones a los parámetros del filtro de Gabor, entonces resultan $M \times N$ mapas de rasgos. Cada MR contiene una serie de proto-objetos [11], término que en [2] lo describen como “unidades volátiles de información visual que se pueden enlazar en un objeto coherente y estable cuando se accede por atención enfocada”, por lo que se emplea una etapa de agrupamiento de datos para determinar en cada MR los pixeles que forman parte de cada proto-objeto, dónde está cada proto-objeto y cuántos proto-objetos existen. Además, la etapa de agrupamiento de datos también es utilizada para discriminar proto-objetos con dimensiones menores a cierto valor propuesto. Posteriormente, se utiliza una red neuronal artificial (RNA) de competencia para discriminar los MRs que contengan un mayor número de proto-objetos, ya que de acuerdo a la definición de atención selectiva pueden contener información irrelevante de la escena. Después de esto, el sistema realiza una discriminación entre los distintos MRs obtenidos para generar un mapa de preponderancias (imagen de salida con los rasgos preponderantes de la escena). La arquitectura del sistema propuesto se muestra en la Figura I.

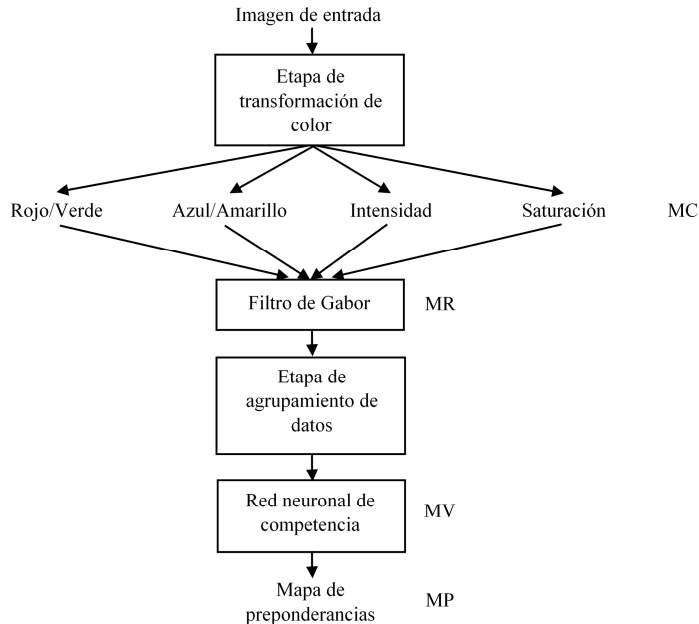


Figura I.- Arquitectura propuesta para el sistema de detección de rasgos preponderantes.

2.1 Etapa de transformación de color.-El proceso que sigue la mente humana para percibir e interpretar el color es un fenómeno fisiológico-psicológico que aún no puede determinarse en su totalidad [9]. El uso de color en el procesamiento de imágenes a menudo resulta de gran ayuda en la identificación y extracción de objetos de una escena. Algunos trabajos enfocados en simular el sistema de visión humano proponen arreglos de mapas de color [2, 5], que oponen distintas componentes de color. Tomando como base esos trabajos, a continuación se describen las ecuaciones que se implementarán para crear los mapas de color:

$$RG = (R' - G')/2, \quad (1)$$

$$BY = (R' + G' - 2B')/4, \quad (2)$$

$$Int = (R' + G' + B')/3, \quad (3)$$

$$Sat = [\max(R', G', B') - \min(R', G', B')]/2. \quad (4)$$

Donde R' , G' y $B' \in [-1,1]$ son las componentes normalizadas en R (rojo), G (verde) y B (azul), por sus siglas en Inglés, de la imagen de entrada; RG es el mapa de color (MC) rojo/verde, BY es el MC azul/amarillo, Int es el MC de intensidad y Sat el MC de saturación. El arreglo propuesto opera a partir de los modelos de color RGB, CBY y HSI.

La entrada a esta etapa es una imagen en formato de color RGB de x número de filas y y número de columnas (recordando que una imagen a color está compuesta por 3 componentes o matrices de datos). A la salida de esta etapa se obtienen 4 mapas de color M (4 matrices de $x \times y$ dimensiones).

2.2 Etapa de extracción de rasgos.- Esta etapa está compuesta por un filtro de Gabor en 2 dimensiones (usualmente utilizada para el procesamiento de imágenes). El filtro de Gabor se obtiene a partir de la multiplicación de una función sinusoidal con una función Gaussiana. La Figura II muestra el filtro de Gabor para señales unidimensionales y la Figura III muestra el filtro de Gabor en 2 dimensiones. En la literatura se han relacionado los procesos de la corteza visual

de los humanos y de algunos mamíferos con el filtrado de las imágenes utilizando funciones de Gabor [6, 12, 13], aplicándose por ejemplo, a la extracción de rasgos de textura en imágenes [14].

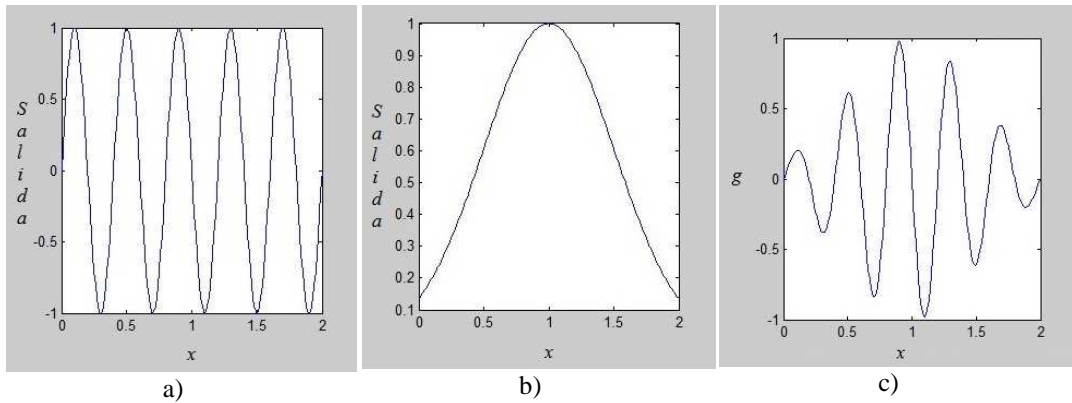


Figura II.- Filtro de Gabor para señales unidimensionales. a) Función sinusoidal. b) Función Gaussiana. c) Filtro de Gabor que resulta de la multiplicación de una función sinusoidal con una función Gaussiana.

El filtro de Gabor en dos dimensiones está compuesto por las siguientes ecuaciones:

$$g(x, y; \theta, \gamma, \sigma, \lambda, \psi) = e^{-\frac{(x'^2 + y'^2)}{2\sigma^2}} \cos\left(2\pi \frac{x'}{\lambda} + \psi\right), \tag{5}$$

$$x' = x \cos\theta + y \sin\theta, \tag{6}$$

$$y' = y \cos\theta - x \sin\theta, \tag{7}$$

donde λ representa la longitud de onda de la señal sinusoidal, θ es la orientación de la normal a las franjas paralelas de una función de Gabor, ψ es el valor de desfaseamiento, σ es el valor de varianza de la función Gaussiana, γ es la razón de aspecto espacial y las variables x y y representan la posición fila-columna de un pixel de la imagen.

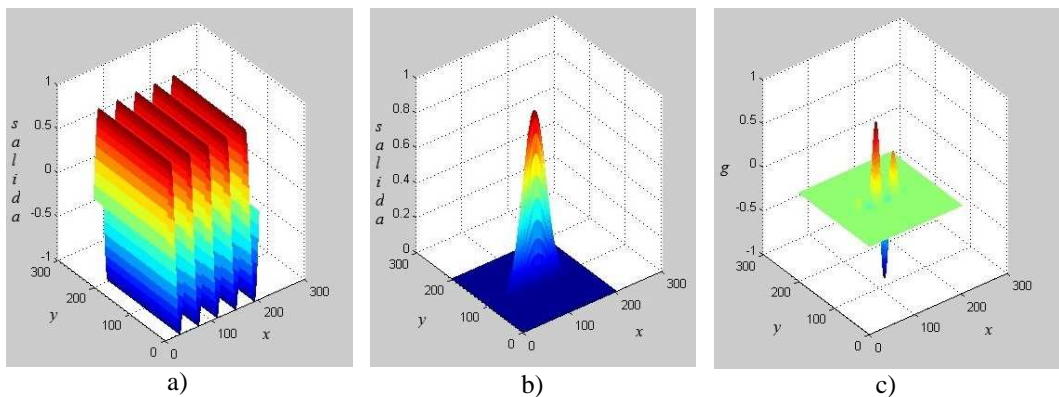


Figura III. Filtro de Gabor en 2 dimensiones (2D). a) Función sinusoidal. b) Función Gaussiana. c) Filtro de Gabor en 2 dimensiones (2D).

En esta etapa de extracción de rasgos ingresan 4 ($M = 4$) mapas de color, a cada uno de ellos se le aplica la función de Gabor, a la cual se le realizan 32 ($N = 32$) variaciones de los distintos

parámetros (explicado a detalle en la sección 3), entonces se obtienen $M \times N = 4 \times 32 = 128$ mapas de rasgos.

2.3 Etapa de detección de rasgos preponderantes basada en objetos.- Esta etapa consta de algoritmos catalogados en el área de cómputo inteligente (soft computing) [7], tales como agrupamiento de datos y redes neuronales artificiales (RNA) [7, 8]. El objetivo de esta etapa es poder discernir información relevante a partir de los mapas de rasgos, es decir, realizar la extracción de patrones.

Como se mencionó en el apartado 2, cada mapa de rasgos muestra una serie de proto-objetos y en [5] se propone una etapa de umbralización. Suponga que el n -ésimo mapa de rasgos del m -ésimo mapa de color se denota por $MR_{m,n}(x,y)$, entonces el conjunto de proto-objetos $Pob_{m,n}$ en $MR_{m,n}$ está dado por:

$$Pob_{m,n} = \{(x,y) | MR_{m,n}(x,y) > \alpha \cdot \max(MR_{m,n}(x,y))\}, \quad (8)$$

donde proponen un valor típico de umbralización $\alpha = 0.6$. Se menciona que a partir de la conexión de las regiones, $Pob_{m,n}$ puede dividirse en $A_{m,n}$ áreas de proto-objetos, denotadas por S_i , donde $i = 1, 2, \dots, A_{m,n}$. Sin embargo, no mencionan el método que utilizan para determinar qué píxeles conforman a cada proto-objeto. En este artículo se propone una etapa de agrupamiento de datos para determinar los píxeles que conforman a cada proto-objeto, dónde está cada uno de ellos y cuántos son.

2.3.1 Etapa de detección de proto-objetos basada en algoritmos de agrupamiento de datos.- Se han desarrollado distintos métodos para la extracción de regiones en imágenes [15], pudiéndose también utilizar redes neuronales artificiales de competencia o bien agrupamiento de datos. Con respecto a las RNAs de competencia utilizadas para el agrupamiento de datos, el tipo de entrenamiento es no supervisado [8] y el número de cúmulos de datos depende del número de neuronas que se pongan a competir. Por otro lado, los algoritmos de agrupamiento de datos se basan en el cálculo de la densidad de los datos, por lo que pueden proporcionar información de cuántos cúmulos existen (dato importante cuando se trabaja en alta dimensión y el diseñador no puede visualizar esa información o bien en sistemas autónomos, como lo es en este caso), además de que el algoritmo permite despreciar los cúmulos que estén con una densidad de datos poco significativa (este valor dependerá de la aplicación). Dichas características dan cierta ventaja con respecto a otros métodos de extracción de regiones ya que sería necesario agregar otras etapas para poder determinar qué regiones resaltadas cumplen con los criterios (tamaño) para ser considerados proto-objetos.

Existen diferentes algoritmos para implementar el agrupamiento de datos [7]. Este trabajo propone la utilización del algoritmo de montaña [16] como método inicial para encontrar la posición (centros iniciales de cada cúmulo) y el número de cúmulos de la distribución de datos y después aplicar el algoritmo K-Means para aumentar la precisión del sistema. Se ha reportado que puede mejorarse el desempeño del sistema si se realizan combinaciones entre algunos de los métodos de agrupamiento de datos [17, 18]. El algoritmo de montaña basa su funcionamiento en sobreponer una rejilla en el espacio dimensional donde se encuentren los datos, siendo el diseñador quien determina la densidad de intersecciones de líneas rectas que tendrá dicha rejilla dependiendo de la aplicación (ver ejemplo en Figura IV). Cada intersección de la rejilla se considera un centro de cúmulo potencial y el algoritmo obtiene la densidad de datos para cada uno de ellos. Existe una relación entre la densidad de intersecciones y la precisión del algoritmo para encontrar los centros de los cúmulos; claro que a mayor densidad de intersecciones aumenta el gasto computacional. Con respecto al algoritmo K-Means, este realiza una clasificación certera (para cada dato asigna grados de pertenencia de 0 o 1 en cada cúmulo). La suma de grados de pertenencia de un dato en el total de cúmulos debe ser igual a 1.

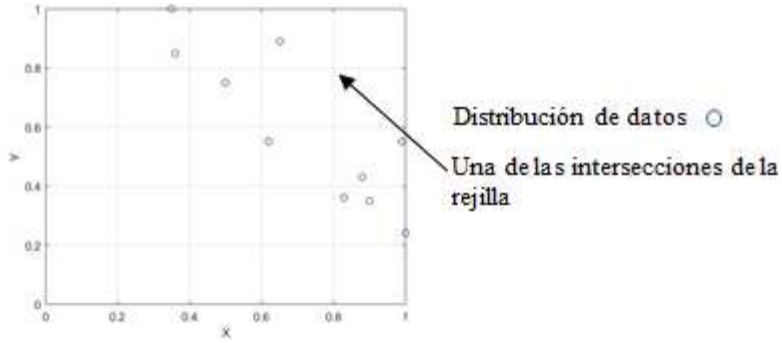


Figura IV. Ejemplo de una distribución de datos en 2 dimensiones (círculos) y una rejilla superpuesta con incrementos de 0.2 en el rango $[0,1]$ de los ejes x y y [16]. Las intersecciones de la rejilla serán centros de cúmulos potenciales.

Por lo tanto, el algoritmo de montaña determinará el número de cúmulos (proto-objetos) dentro de un MR y la posición de los centros de cada cúmulo. Al aplicar posteriormente el algoritmo K-Means, se aumenta la precisión de la posición de los centros de cada cúmulo, parte relevante a la hora de realizar la asignación o clasificación de cada dato (determinar qué píxeles conforman a cada proto-objeto).

Como se mencionó anteriormente, el algoritmo de montaña basa su funcionamiento en sobreponer una rejilla de líneas rectas en el espacio dimensional de los datos a agrupar (clasificar). Cada intersección de las líneas de la rejilla representa un centro de cúmulo potencial. Considere que el universo de datos U está dado por:

$$U = \{\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k\}, \quad (9)$$

donde \hat{u} representa un vector de datos con un número de elementos igual al espacio dimensional donde se esté trabajando, k representa el número total de vectores (para nuestro caso cada vector de dato representa la posición fila-columna de un pixel de un mapa de rasgos). El siguiente paso es construir una función montaña que representa la medida de densidad de los datos. La altura de la función montaña en un punto $\hat{v}_j \in V$, donde V es el total de intersecciones de la rejilla propuesta está dado por:

$$\text{Montaña}^1(\hat{v}_j) = \sum_{i=1}^k e^{(-\vartheta \cdot d(\hat{u}_i, \hat{v}_j))}, \quad (10)$$

donde \hat{u}_i es el i -ésimo punto de dato (vector de dato), ϑ es una constante positiva y $d(\hat{u}_i, \hat{v}_j)$ es la distancia Euclidiana entre el i -ésimo punto de dato y la j -ésima intersección de la rejilla. Cada punto de dato \hat{u}_i contribuye a la altura de la función montaña en el punto \hat{v}_j (j -ésima intersección de la rejilla) y esa contribución es inversamente proporcional a la distancia entre \hat{u}_i y \hat{v}_j . El superíndice en $\text{Montaña}^1(\hat{v}_j)$ indica que es la primera montaña obtenida por el algoritmo. Para obtenerla es necesario realizar un barrido de todos los puntos de intersección \hat{v}_j aplicando (10). La primera montaña obtenida corresponde a la mayor densidad de datos, en nuestro caso, el proto-objeto con mayor área.

El primer centro de cúmulo está definido por:

$$\hat{c}_1 = \max[\text{Montaña}^1(\hat{v}_j)]. \quad (11)$$

El siguiente paso consiste en encontrar los siguientes cúmulos o montañas, por lo que debe eliminarse la primera montaña encontrada para que no interfiera en los nuevos cálculos. Esto se logra aplicando la siguiente ecuación:

$$Montaña^q(\hat{v}_j) = \max \left[Montaña^{q-1}(\hat{v}_j) - \hat{c}_{q-1} \sum_{q=1}^Q e^{(-\tau \cdot d(\hat{c}_{q-1}, \hat{v}_j))}, 0 \right], \text{ para } q > 1. \quad (12)$$

donde τ es una constante. La ecuación (12) realiza una sustracción a cada nodo de la montaña recién encontrada; el valor sustraído en cada nodo es proporcional a la distancia de ese punto hacia el centro de la montaña anterior (\hat{c}_{q-1}). El paso anterior se repite hasta encontrar el número de montañas que se requieran o bien puede utilizarse una condición de paro, pudiéndose utilizar la razón entre el valor máximo de la primera montaña \hat{c}_1 y el valor máximo de la montaña previa \hat{c}_{q-1} :

$$\frac{\hat{c}_1}{\hat{c}_{q-1}} < \varepsilon, \quad (13)$$

donde ε es un parámetro de paro definido por el usuario.

El algoritmo de montaña proporciona la posición de los centros de cada cúmulo y también puede indicar la pertenencia de cada punto de dato en cada cúmulo. En este trabajo se utiliza el algoritmo de montaña para generar la cantidad de cúmulos existente en las distribuciones de datos (mapas de rasgos) y la posición de los centros de cúmulos. Esta información sirve como parámetros de entrada al algoritmo K-Means que se encarga de mejorar la precisión de los centros de los cúmulos y a partir de eso determinar a qué cúmulo pertenece cada pixel del MR. El algoritmo K-Means puede operar por sí solo, pero se tiene el inconveniente de que el algoritmo no puede determinar el número de cúmulos que existe en el espacio dimensional.

Al igual que en el algoritmo de montaña, el algoritmo K-Means particiona una distribución de k puntos de dato dentro de un universo U (9) en Q grupos o regiones. El objetivo del algoritmo es asignar o clasificar cada punto de dato en un determinado cúmulo a partir del cálculo de la menor distancia Euclidiana entre los puntos de dato en el universo U con respecto a los centros candidatos. Cada punto será asignado a uno y solo un cúmulo o partición a partir de la siguiente función característica:

$$\chi_{\text{cúmulo}_q}(\hat{u}_i) = \begin{cases} 1 & \hat{u}_i \in \text{cúmulo}_q \\ 0 & \hat{u}_i \notin \text{cúmulo}_q \end{cases}. \quad (14)$$

Una asignación de pertenencia donde a cada punto de dato \hat{u}_i existente dentro del espacio U se le asigna uno de los centros \hat{c}_q candidatos puede definirse como:

$$\chi_{iq} = (i = 1, 2, \dots, k; q = 1, 2, \dots, Q), \quad (15)$$

Creándose con esa partición una matriz de pertenencia dada por:

$$Matriz_q = \{U | \chi_{iq} \in \{0, 1\}, \sum_{q=1}^Q \chi_{iq} = 1, 0 < \sum_{i=1}^k \chi_{iq} < k\}, \quad (16)$$

donde U es una matriz de q filas y k columnas. El algoritmo propuesto para las particiones es una aproximación a partir de la suma de los errores cuadráticos empleando distancia Euclidiana entre dos puntos. La función costo o función objetivo se define entonces como:

$$J(U, \hat{c}) = \sum_{i=1}^k \sum_{q=1}^Q \chi_{iq} (d_{iq})^2, \quad (17)$$

donde \hat{c} es un vector que contiene los centros de cúmulo y d_{iq} es la distancia Euclidiana medida en un espacio de p dimensiones entre el i -ésimo punto de dato y el q -ésimo centro, descrita de la forma:

$$d_{iq} = d(\hat{u}_i - \hat{c}_q) = \|\hat{u}_i - \hat{c}_q\| = \left[\sum_{j=1}^p (\hat{u}_{ij} - \hat{c}_{qj})^2 \right]^{1/2}. \quad (18)$$

Los p elementos del vector \hat{c}_q se calculan a partir de:

$$\hat{c}_{qj} = \frac{\sum_{i=1}^k \chi_{iq} \cdot \chi_{qj}}{\sum_{i=1}^k \chi_{iq}}. \quad (19)$$

Para el criterio de convergencia puede emplearse la variación de la posición de los centros de cúmulos como condición de paro (si todos los centros de cúmulo varían de una iteración a otra dentro de cierto rango el algoritmo se detiene) o lo que es el mismo concepto, que la asignación de los puntos de datos entre iteraciones ya no presente cambios.

2.3.2 Etapa de competencia.- Una vez establecidos los proto-objetos en cada mapa de rasgos se propone utilizar una RNA de competencia para realizar un proceso de inhibición entre los distintos mapas de rasgos y así poder modelar la atención selectiva, tomando como premisa que un mapa de rasgos con mayor cantidad de proto-objetos y/o áreas grandes de esos proto-objetos contiene mayor cantidad de información irrelevante, caso contrario a mapas de rasgos con pocos proto-objetos y áreas moderadas. La ecuación utilizada para ponderar la información de cada MR con respecto a la sumatoria de áreas de todos los proto-objetos es una función sigmoïdal [7] acotada en $0 \leq A_{m,n} \leq 1$:

$$A_{m,n} = \frac{1}{1+e^{[\omega \cdot (r_{m,n})]}} \quad (20)$$

donde ω es un parámetro que determina la curva y $r_{m,n}$ es la razón entre la sumatoria de las áreas de todos los proto-objetos y el área total del $MR_{m,n}$. Debido al uso de los algoritmos de agrupamiento de datos se puede conocer el número de píxeles que conforman a cada proto-objeto, por lo que se ha utilizado este dato como valor de área. Si en (20) se utiliza un valor de $\omega < 0$, el valor de $A_{m,n}$ será proporcional al valor de $r_{m,n}$, es decir, a mayor valor de la sumatoria de áreas de los proto-objetos, mayor será el valor de $A_{m,n}$.

La siguiente ecuación genera mapas de visibilidad (MV), es decir, cada MR es dividido entre la sumatoria de áreas ponderada y la cantidad de proto-objetos que contiene, por lo que MRs con mayor número de proto-objetos y/o mayor área de éstos, tendrán un menor valor de visibilidad:

$$MV_{m,n}(x, y) = \frac{MR_{m,n}(x,y)}{A_{m,n} \cdot Q_{m,n}} \quad (21)$$

donde $Q_{m,n}$ es el número de proto-objetos dentro del $MR_{m,n}$.

Para la obtención del mapa de preponderancias únicamente se elige el mapa de visibilidad con mayor valor. El mapa de preponderancias es la salida del sistema.

$$MP(x, y) = \max\{MV_{m,n}(x, y)\}, \quad m = 1, \dots, M; n = 1, \dots, N. \quad (22)$$

3. Resultados.- Para las pruebas del sistema se utilizaron imágenes de escenas naturales con dimensiones de 640×480 píxeles (formato VGA) y en formato de color RGB. Con respecto al filtro de Gabor, como se mencionó en el capítulo anterior, está definido por 7 parámetros, de los cuales 2 son las coordenadas en los ejes x, y del filtro. Otro aspecto importante es el tamaño del filtro de Gabor. Los parámetros utilizados fueron: *Tamaño de filtro* = 13×13 píxeles, $\lambda = 6$, $\theta = 0: \pi/8 : 7\pi/8$, $\psi = 0$, $\gamma = 0.4: 0.2: 1$ y $b = 4$. Resultando en 32 variaciones de los parámetros del filtro de Gabor (8 rotaciones para θ y 4 variaciones de la longitud de onda sinusoidal γ). Debido a que el sistema aplica el filtro de Gabor a cada uno de los 4 mapas de color MC, se generan 128 mapas de rasgos MR (salida de la etapa que aplica el filtro de Gabor). Con respecto al algoritmo de montaña, se utilizaron los siguientes valores: $\vartheta = 1.5$, $\tau = 0.01$ y $\varepsilon = 1$. Para la etapa de competencia se utilizó $\omega = -1.5$.

La Figura V muestra una imagen de prueba y los correspondientes mapas de color (MC) resultantes después de pasar por la etapa de transformación de color. En la Figura VI se presenta la salida de cada una de las etapas del sistema para uno de los 128 mapas de rasgos (MR). En la Figura VI.c se muestran los cúmulos y centros de cúmulos (puntos más oscuros) encontrados para ese MR y la Figura VI.d muestra el mapa de preponderancias, que es la salida total del sistema.

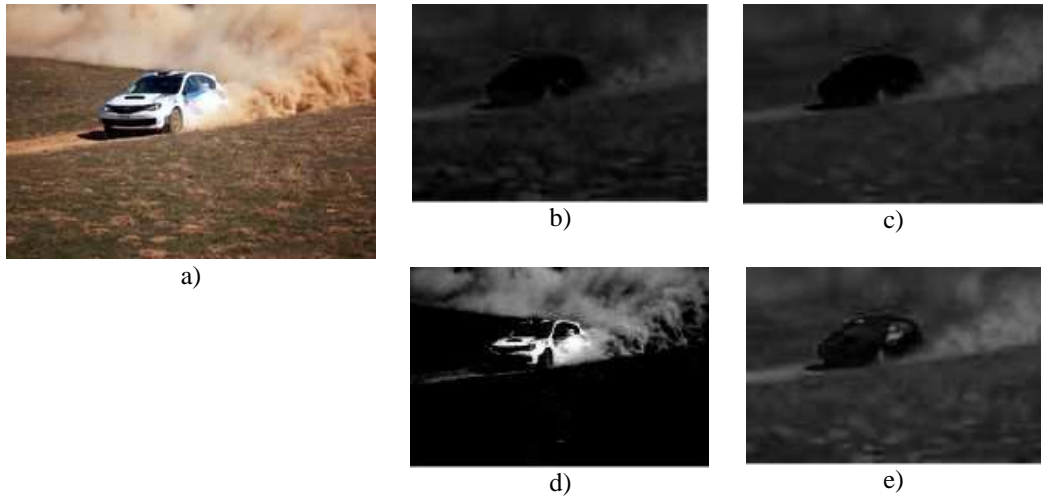


Figura V. a) Imagen de prueba. b) Mapa de color Rojo/Verde. c) Mapa de color Azul/Amarillo. d) Mapa de color de Intensidad. e) Mapa de color de Saturación.

La Tabla I muestra una serie de imágenes de prueba del sistema, los mapas de preponderancias generados por el sistema propuesto y los tiempos de procesamiento obtenidos al ejecutar el algoritmo con un procesador Intel Core i5-6200U a 2.4GHz, con 8GB de memoria RAM y sistema operativo de 64 bits. También se incluyen los mapas de preponderancias obtenidos mediante la herramienta Saliency Toolbox 2.3 [2], con fines de comparación de los rasgos preponderantes que genera cada sistema. Como se mencionó al principio de esta sección, todas las imágenes son de 640×480 píxeles, con excepción de la figura de Lena (imagen 9 de la Tabla I), que tiene una dimensión de 256×256 píxeles.

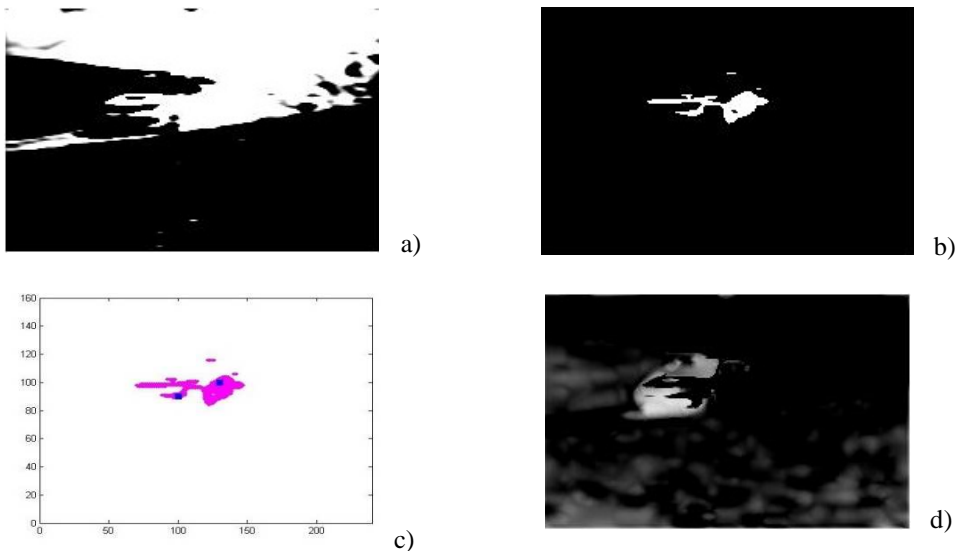

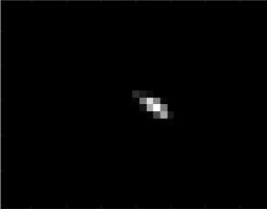
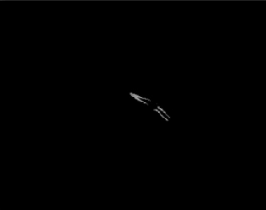

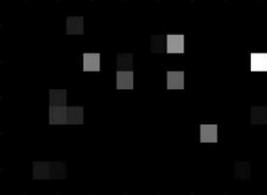


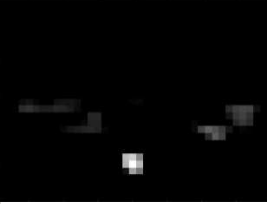


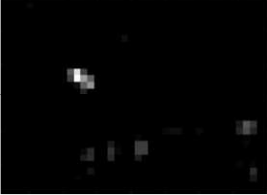


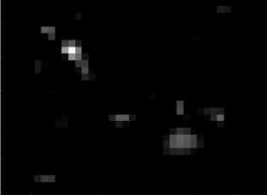


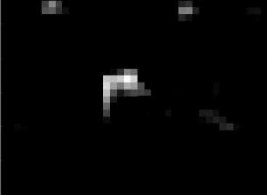



Figura VI. Salida de cada una de las etapas del sistema para uno de los 128 MRs generados. a) Salida de la etapa del filtro de Gabor. b) Imagen resultante después de umbralizar la imagen en (a). c) Salida de la etapa de agrupamiento de datos. d) Imagen de salida del sistema (mapa de preponderancias).

Número de imagen <i>n</i>	Imagen de entrada	MP generado por modelo de Walther y Koch	MP obtenido por el sistema propuesto	Tiempo de procesamiento [s]
1				274.15
2				213.55
3				173.76
4				861.23
5				1247.87
6				350.77

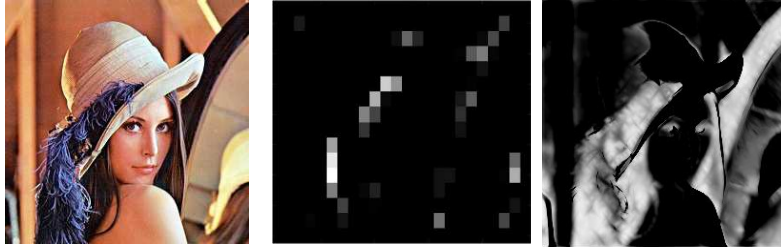


Tabla I.- Imágenes de prueba del sistema, mapas de preponderancias obtenidos por el modelo de Walther y Koch, por el modelo propuesto en este trabajo y los tiempos de procesamiento del modelo propuesto.

4. Conclusiones.- En este trabajo se ha abordado el proceso de obtención de rasgos preponderantes en imágenes de escena natural, tomando como base un modelo de atención visual ascendente. El sistema posee una etapa de transformación de color, que combina distintas componentes de los formatos de color RGB, HSI y CMY para generar 4 matrices de color a partir de la imagen de entrada, lo cual se puede traducir como dotar al sistema de un mayor número de características para realizar una mejor discriminación de información, sin embargo, al aumentar la cantidad de información, se aumenta el costo computacional. Con respecto al filtro de Gabor, demostró ser un filtro robusto para la extracción de características de las imágenes; dependiendo del número de variaciones de sus parámetros y de las dimensiones del filtro puede generar mayor o menor detalle de las características de las imágenes, pero de nuevo, a mayor detalle mayor será el costo computacional.

El sistema modela la atención selectiva a partir de la implementación de algoritmos de agrupamiento de datos y de una red neuronal artificial de competencia que opera a partir del área y número de objetos (proto-objetos) encontrados en cada mapa de rasgos. El tiempo de procesamiento para cada imagen varía en función de la complejidad de la escena, que a su vez será proporcional al número de proto-objetos que se encuentren; esta etapa del sistema utiliza algoritmos inteligentes que se adaptan a la información que ingresa, por lo que pueden observarse tiempos muy distintos para imágenes de iguales dimensiones.

Para las pruebas, fueron utilizadas imágenes con escenas de distinta complejidad. Para imágenes sencillas el sistema genera los rasgos preponderantes en las mismas regiones que el modelo de Walther y Koch, pero se obtiene mayor detalle de los rasgos encontrados. En la escena de las abejas (ver Tabla I, imagen 2) el sistema realiza una discriminación total del panal, mientras que el modelo de Walther y Koch presenta activaciones en algunas regiones del panal. Para imágenes con escenas de mayor complejidad, el sistema genera más regiones de interés en comparación con el modelo de Walther y Koch. Si el objetivo del sistema es que opere como una etapa previa al reconocimiento de patrones, puede verse como una ventaja que el sistema arroje más detalles de las regiones preponderantes, ya que facilitará el reconocimiento. Sin embargo, si el sistema propuesto se requiere únicamente para localizar puntos de atención en la escena, a mayor complejidad de la escena, la cantidad de regiones de interés que genera (comparado con el modelo de Walther y Koch) puede resultar en una desventaja. Es posible disminuir el grado de detalle del sistema variando algunos de los parámetros en la etapa de aplicación del filtro de Gabor y/o en la etapa de agrupamiento de datos pero al aumentar la selectividad en escenas complejas, puede afectarse el desempeño en imágenes sencillas. El sistema propuesto puede catalogarse como una etapa de extracción de rasgos, es decir, está eliminando información no relevante y resaltando los rasgos preponderantes de la escena visual (regiones de interés), por lo que puede ser acoplado a etapas posteriores de reconocimiento de patrones.

5. Agradecimientos.- Los autores agradecen a la Secretaría de Investigación y Posgrado (SIP) del Instituto Politécnico Nacional por el apoyo brindado en el desarrollo de este trabajo.

6. Referencias

- [1] Frintrop S., Rome E., Christensen H. I. *Computational visual attention systems and their cognitive foundations: a survey*. ACM Trans. Applied Perception, 2010. Vol. 7, No. 1, Article 6.
- [2] Walther D., Koch C. *Modeling attention to salient proto-objects*. 2006. Neural Networks, 19: 1395-1407.
- [3] Borji A., Itti L. *State-of-the-art in visual attention modeling*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013. Vol. 35, Issue: 1: 185-207.
- [4] Itti L., Koch C., Niebur E. *A model of saliency-based visual attention for rapid scene analysis*, 1998. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11):1254-1259.
- [5] Chen D., Wu C. *Object-based multi-feature competitive model for visual saliency detection*. Proceedings of the Second International Conference on Intelligent System Design and Engineering Application. 2012: 1079-1082.
- [6] Petkov N. *Biologically motivated computationally intensive approaches to image pattern recognition*. Future Generation Computer Systems, 1995. 11 (4-5): 451-465.
- [7] Jang J., Sun C., Mizutani E. *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*, 1997, 1st Edition. USA: Prentice Hall.
- [8] Hagan M., Demuth H., Beale M. *Neural network design*, 2002, 1ª edición. Pws Publishing Company.
- [9] Gonzalez R., Woods R. *Digital Image Processing*, 2008, 3th edition. Pearson International Edition.
- [10] Walther D., Rutishauser U., Koch C., Perona P. *Selective visual attention enables learning and recognition of multiple objects in cluttered scenes*. Computer vision and Image Understanding, 2005. Vol. 100 (1-2): 41-63.
- [11] Rensink R. *Seeing, sensing and scrutinizing*. Vision research, 2000. Vol. 40 (10-12): 1469-1487.
- [12] Marcelja S. *Mathematical description of the response of simple cortical cells*. J. Opt. Soc. Am., 1980. Vol. 70, No. 11: 1297-1300.
- [13] Daugman J. *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters*. Journal of the Optical Society of America A, 1985. 2 (7):1160-1169.
- [14] Kruijzinga P., Petrov N., Grigorescu S. E. *Comparison of texture features based on Gabor filters*. IEEE Transactions on Image Processing, 2002. Vol. 11, No. 10: 1160-1167.
- [15] Pajarez G., de la Cruz J. *Visión por computador. Imágenes digitales y aplicaciones*, 2008, 2ª edición. Alfaomega.
- [16] Yager R., Filev D. *Approximate clustering via the mountain method*. IEEE Transactions on Systems, Man and Cybernetics, 1994. Vol. 24, No. 8.
- [17] Guevara E., Sánchez O., González Y. *Segmentación de imágenes empleando el método de la montaña y sustractivo para la aceleración del procesamiento de los algoritmos K-Means y C-Means*. Boletín UPIITA, No. 51.
<http://www.boletin.upiita.ipn.mx/index.php/component/content/article/9-articles/23-numeros-antiores-cyt> Fecha de último acceso: 03/05/2017.
- [18] Guevara E., Sánchez O., González Y. *Segmentación de Imágenes a Color mediante Técnicas de Agrupamiento de Datos empleando los Algoritmos K-Means y C-Means*. Boletín UPIITA, No. 51.
<http://www.boletin.upiita.ipn.mx/index.php/component/content/article/9-articles/23-numeros-antiores-cyt> Fecha de último acceso: 03/05/2017.