

Localización Automática de Objetos 2D y3D en Imágenes

Automatic 2-D and 3-D Object Localization on Images

Raúl Pinto Elías¹ y Humberto Sossa²

¹ Centro Nacional de Investigación y Desarrollos Tecnológico
Prolongación Av. Palmira s/n C.P. 62051, Apdo. Postal 5-377
Cuernavaca Morales, México

² Centro de Investigación en Computación-IPN
Av. Juan de Dios Bátiz esquina con M. Othón de Mendizabal
Colonia Nueva Industrial Vallejo, México, D. F. 07738

E-mails: rpinto@infosel.net.mx, rpinto@cenidet.edu.mx, hsossa@cic.ipn.mx

Artículo recibido en Junio 25, 2001; aceptado en Febrero 19, 2002

Resumen

Se presenta una nueva técnica para la detección automática de objetos a partir de una imagen. Inicialmente se divide la imagen en pequeñas celdas de tamaño fijo. A continuación, se calculan los invariantes de Hu para cada celda (Hu 62).

La detección de objetos se lleva a cabo por el agrupamiento de celdas individuales usando un algoritmo genético al ajustar una función de costo dada. Esta función corresponde a la descripción invariante del objeto dado en términos de los valores de sus niveles de gris asociados. La técnica fue probada en dos escenarios. Inicialmente con objetos simples tales como caracteres impresos y posteriormente con objetos más complicados como componentes de rostros humanos, ojos, nariz y boca.

Palabras clave: Reconocimiento de objetos, Reconocimiento Invariante de Objetos, Invariantes Geométricos, Algoritmo Genético.

Abstract

We present a new technique for the automatic detection of objects from in an image. At the beginning the image is divided into small cells of fixed size. Next Hu invariants (Hu 62) are computed for each cell. Grouping individual cells using a genetic algorithm by adjusting a given cost function carries out object detection. This function corresponds to an invariant description of the object in terms of the gray-level values of its pixels. The technique was tested on two scenarios. Initially with very simple objects such as printed characters and then with more complicated objects such as face components: eyes, nose and mouth.

Keywords: Object Recognition, Invariant Object Recognition, Geometric Invariants, Genetic Algorithm.

1 Introducción

Un problema básico en visión por computadora es el reconocimiento de objetos. Su solución es crucial en muchas aplicaciones como indexado de vídeo en tiempo real y el guiado automático de robots. Aun en el caso más simple, de reconocimiento en dos dimensiones, el problema no ha sido completamente resuelto. Muchas soluciones a este problema han sido propuestas hasta ahora, ver por ejemplo (Guzmán, 68; Dudani, 77; Perkins, 78, Davis, 79; Bhanu, 84; Ayache, 86; Sossa, 92; Reiss, 93; Jacobsen, 93, Rothwell, 95 and Sossa, 97, Sossa, 98).

En este trabajo se presenta una técnica completamente diferente. A través de ella se pretende realizar el proceso de segmentación y reconocimiento al mismo tiempo, mediante un proceso que considera y evalúa más de una región candidato a la vez. La técnica se vale de un algoritmo genético y una descripción invariante del objeto para llevar a cabo la tarea especificada. Los descriptores usados son los bien conocidos invariantes a traslaciones, rotaciones y cambios de escala propuestos por Hu.

En una primera etapa, una imagen posiblemente conteniendo un objeto de interés, es dividida en pequeñas celdas de tamaño fijo. A continuación los momentos ordinarios se calculan para todas y cada una de las celdas.

A partir de estas cantidades se obtienen los correspondientes invariantes de Hu. La detección de objetos se lleva a cabo a través del agrupamiento de celdas individuales usando un algoritmo genético por el ajuste de una función de costo específica. Esta función involucra a la descripción invariante del objeto dado en términos de los valores de niveles de gris asociados. Inicialmente, el método fue probado en la detección de objetos simples tales como

caracteres impresos; posteriormente, el método fue probado para detectar objetos más complejos como componentes de rostros humanos, ojos, nariz y boca.

El resto de este trabajo está organizado como sigue. En la segunda sección se describen cada una de las etapas de la técnica propuesta. En la tercera sección se presentan los rasgos usados para describir un objeto, mientras que en la sección cuatro se describe el algoritmo de búsqueda usado para detectar un objeto dado. En la sección cinco se muestran algunos resultados experimentales donde se prueba dicha técnica. Finalmente, en la sexta sección se dan las conclusiones y algunas directivas de trabajo futuro.

2 La Técnica

La técnica propuesta se compone de tres etapas. Cada una de estas etapas se explica a continuación.

Representación de los objetos. En el contexto de este trabajo, un objeto es representado como un patrón, que es la traducción literal del término anglosajón *pattern* (González, 96). Como tal, dicho objeto es descrito como un vector de rasgos invariantes como sigue:

$$O_i = \{\Phi_1, \Phi_2, \dots, \Phi_K\} \quad (1)$$

con

$$\Phi_k = f_k(O_i), 1 \leq i \leq n; \quad 1 \leq k \leq K \quad (2)$$

donde:

- O_i es el i -ésimo objeto que se describe.
- Φ_k es el k -ésimo descriptor.
- $f_k(O_i)$ es la función generadora del k -ésimo descriptor.
- n es el número de objetos.
- K es el número de descriptores.

Definición de la función de costo. Sea esta función $C(f_k)$. El objetivo de esta función es guiar el proceso de búsqueda de la descripción del objeto en el espacio de representación E . Para una imagen dada, E se compone de la unión de todos los posibles valores de f_k ,

$$E = \bigcup_{r=1}^R f_k(r), k = 1, 2, \dots, K \quad (3)$$

donde R es el número total de regiones disjuntas de pixeles en que una imagen particular puede ser dividida para encontrar un objeto dado. Un análisis de los dos casos extremos relacionados con la cardinalidad de E (el mejor y el peor caso) se presenta en el Apéndice A.

Proceso de búsqueda. Este proceso se realiza mediante el recorrido del espacio de representación E por medio de un algoritmo de búsqueda dirigida hasta encontrar, dentro de E , la mejor aproximación de una f_k dada. El mecanismo de búsqueda recorre a E mediante la fusión y separación de regiones de tamaño variable, que son múltiplos de las celdas ya mencionadas, en la imagen. Esto se hará como veremos en

la Sección 4 usando un algoritmo genético. El resultado final será una región encerrando el objeto de interés.

3 Descriptores Invariantes Usados

Muchos descriptores han sido propuestos en el pasado para describir un objeto. Los descriptores usados en este trabajo son los momentos invariantes a traslaciones, rotaciones y cambios de escala propuestos por Hu (Hu, 62). En este trabajo, se usan únicamente los primeros cuatro invariantes.

De acuerdo a la Sección 2, un objeto puede ser descrito como:

$$O_i = \{\phi_1, \phi_2, \phi_3, \phi_4\} \quad (4)$$

con $1 \leq i \leq n$, y es n el número de objetos. Sólo para recordar, los cuatro primeros invariantes de Hu pueden calcularse como sigue:

$$\phi_1 = \mu_{20} + \mu_{02} \quad (5)$$

$$\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (6)$$

$$\phi_3 = (\mu_{30} + 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \quad (7)$$

$$\phi_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \quad (8)$$

donde los μ_{pq} , son los momentos centrales invariantes a traslaciones obtenidos a la vez en términos de los momentos ordinarios, m_{pq} (González, 96). Recordemos que ambos momentos de orden $(p+q)$ para una función discreta de dos variables $f(x,y)$ se definen como:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (9)$$

y

$$\mu_{pq} = \sum_x \sum_y (x^p - \bar{x})(y^q - \bar{y}) f(x, y) \quad (10)$$

con

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{y} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (11)$$

Para poder ser usados en el contexto de este trabajo, los ϕ_i 's deben ser descritos en función de los momentos ordinarios y no en términos de los momentos centrales. Por ejemplo, el primer invariante de Hu, ϕ_1 puede ser expresado como:

$$\begin{aligned} \phi_1 &= m_{20} - \bar{x}m_{10} + m_{02} - \bar{y}m_{01} \\ &= \frac{(m_{20} + m_{02})m_{00} - (m_{10}^2 + m_{01}^2)}{m_{00}} \end{aligned} \quad (12)$$

La razón es que los momentos ordinarios pueden ser sumados directamente ya que están referidos al mismo punto (el origen del marco de referencia). Los momentos centrales, por el contrario, se encuentran referidos a puntos diferentes (los centroides de cada celda).

Si quisiéramos usar los momentos invariantes de cada región directamente para obtener los momentos invariantes de regiones más grandes, sería entonces necesario volver a calcular primeramente los centroides de cada celda y entonces los invariantes para estos nuevos centroides.

Si la región de interés R se divide en r sub-regiones R_i , $1 \leq i \leq r$, los momentos de inercia de toda la región se pueden obtener mediante la suma de los momentos sobre cada sub-región como:

$$m_{pq}(R) = \sum_{i=1}^r (m_{pq}(R_i)) \quad (13)$$

Para cada sub-región los momentos ordinarios pueden ser evaluados por medio de la ecuación (9), como:

$$m_{pq}(R_i) = \sum_x \sum_y x^p y^q f(x, y) \quad (14)$$

Supóngase ahora que la imagen completa es dividida en celdas de tamaño fijo usando una retícula como se muestra en la Figura 1(a). Si para cada celda, los momentos ordinarios son calculados usando la ecuación 14, obtendríamos un mapa de momentos. Al momento que esta retícula es aplicada sobre la imagen, un objeto puede ser visto como formado por un subconjunto de las celdas de la retícula, como se muestra en la Figura 1(b). De esta manera, el mismo objeto a diferentes tamaños, incluirá más o menos celdas. Cada una de estas celdas como se dijo anteriormente es descrita como un vector cuyas cuatro componentes son los primeros cuatro invariantes de Hu.

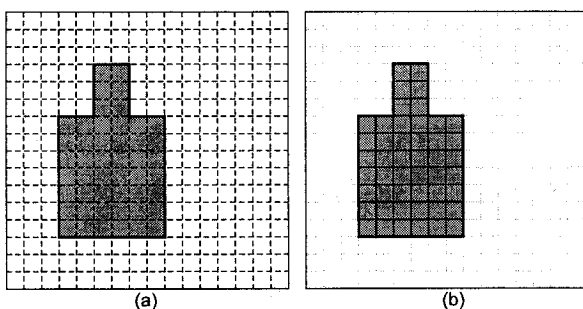


Figura 1. (a) Una imagen dividida en celdas del mismo tamaño al usar una retícula fija. (b) Un objeto de interés subdividido en celdas

El problema del aislamiento de un objeto de una imagen (de su detección en la imagen) puede formularse entonces como un problema de optimización: **escoger de entre el conjunto de celdas en la imagen aquellas contiguas cuyas descripciones locales, al ser sumadas término a término, den como resultado la descripción del objeto buscado.** Se trata pues de optimizar una función, aquella que mejor aproxime la descripción del objeto buscado. La función por aproximar es el vector de rasgos, describiendo el objeto en cuestión, cuyos componentes son los cuatro invariantes de Hu, calculados en términos de las ecuaciones 5 a 8.

4 Algoritmo de Búsqueda Usado

Como se muestra en el Apéndice A, el espacio de posibles soluciones que corresponde al número de posible ensambles de celdas contiguas para formar el objeto buscado es demasiado grande para ser recorrido de manera exhaustiva. Aun en el caso del aislamiento (segmentación) objetos compactos (un rostro, un auto, etcétera), el número restante de soluciones es aún demasiado grande para ser recorrido.

En el contexto de la presente investigación, E se compone de todos los posibles valores que los momentos invariantes de Hu pueden tomar para cada región y para todas las posibles particiones. El resto de esta sección está dedicado a explicar el algoritmo de búsqueda usado para barrer E , sin recorrerlo exhaustivamente.

Si $O_{Interes} = \{\phi_1, \phi_2, \phi_3, \phi_4\}$ corresponde a la descripción del objeto cuya presencia se desea verificar en la imagen, entonces el problema se reduce a encontrar el conjunto de celdas cuyos descriptores invariantes de la región $\{\phi_1, \phi_2, \phi_3, \phi_4\}$ obtenidos al combinar los momentos estándar de cada región de acuerdo a las ecuaciones (5) a (10) se aproximan mejor a los descriptores del objeto de interés $O_{Interes}$.

El agrupamiento de las celdas se realiza mediante un algoritmo genético (AG), el cual permite recorrer el espacio de representación al usar una función de confianza $C(f_k)$. Un AG (Goldberg, 89) ha sido escogido ya que como se ha demostrado en (Pal, 96), los AG's han probado ser muy útiles en muchos problemas incluyendo el reconocimiento de patrones.

Sólo para recordar, en la naturaleza se han desarrollado mecanismos muy potentes que permiten a las diferentes especies evolucionar y adaptarse al medio. Los organismos que son poco aptos para un determinado medio mueren, mientras que los más aptos viven y se reproducen. De esta manera, las nuevas generaciones de organismos son semejantes a sus padres donde los más aptos prevalecen. Estos organismos o especies evolucionan con su entorno, ante cambios pequeños del medio ambiente los organismos se adaptan; sin embargo, los cambios bruscos del medio, por lo general, provocan la desaparición de especies enteras. En la naturaleza se producen mutaciones al azar, donde algunas mueren pronto, mientras que otras pueden producir nuevas especies adaptadas a su entorno.

Un algoritmo genético es un procedimiento computacional que modela los mecanismos de los sistemas genéticos naturales para resolver problemas de optimización. Un AG actúa como una metáfora biológica e intenta imitar algunos de los procesos observados en la evolución natural. Se pueden ver como técnicas de búsqueda y optimización aleatorias estructuradas. Los AG's se ejecutan iterativamente sobre un conjunto de soluciones codificadas, que se conocen como *población*, con tres operadores básicos: *selección/reproducción*, *cruza* y *mutación*.

En el contexto de los AG's, un cromosoma es una posible solución del problema de interés y constituye un elemento de la *población*. Esta posible solución, por lo general, se codifica como una cadena de ceros y unos ("01010..."). Esta representación permite aplicar los operadores, ya mencionados, para combinar dos elementos de la población (padres) y generar nuevos elementos (hijos). A cada cromosoma se le asocia un valor de *aptitud* que indica la utilidad o habilidad del individuo que el cromosoma representa.

La representación de un problema es considerada crucial para el éxito de un AG. Ante esto, uno debe preguntarse si la representación del problema es adecuada para un AG, más que un AG es adecuado para un problema específico (Davior, 91). La representación comúnmente utilizada es la representación binaria (Michalewicz, 92), pero está reportado el uso de otros formatos como los números enteros y reales (Hollnad, 75).

Las principales características del AG usado son las siguientes:

- Un cromosoma equivale a una región rectangular definida por dos puntos $p_1(x1, y1)$ y $p_2(x2, y2)$. Esta región es un múltiplo exacto de las celdas de 5×5 .
- La población inicial se forma con regiones que contienen solamente una celda.
- Las nuevas generaciones se obtienen mediante la recombinación de la última población con la anterior.
- Para la fase inicial de exploración, se considera una probabilidad de cruce del 70% y de mutación del 5%.
- Para la fase de final explotación, se considera una probabilidad de cruce del 90% y de mutación del 1%. En la fase de explotación se aplica, adicionalmente, una optimización local.
- El tamaño de cada población es de 100 y permanece constante.
- Se utilizan múltiples puntos de cruce, en total cuatro puntos fijos.
- La aptitud de un cromosoma se obtiene con una valor de confianza que involucra los invariantes de Hu del objeto de interés y los correspondientes invariantes de la región que se evalúa, así como de un factor de forma.

$$apt = \left| \frac{\phi_i - \phi_d}{\phi_i + \phi_d} \right| * f_f \quad (15)$$

donde:

- Apt es el valor de aptitud del cromosoma.
- ϕ_i es el valor de los invariantes del objeto de interés.
- ϕ_d es el valor de los invariantes de la región de la imagen especificada por un cromosoma.
- f_f es el factor de la forma definido por:

$$f_f = \max \left(\frac{E_M}{E_m}, \frac{E_m}{E_M} \right)$$

donde E_m es el tamaño del eje principal menor y E_M es el tamaño del eje principal mayor. En la figura 2, se muestran los ejes menor y mayor de un objeto.

- El AG procesa un número arbitrario de generaciones, definido en el momento de la simulación. Las simulaciones que se presentan en la Sección 5 se realizaron con 5 generaciones¹.
- La función de costo a minimizar se define como:

$$C(f_k) = \frac{\phi_i - \phi_d}{\phi_i + \phi_d} \quad (16)$$

donde ϕ_i , ϕ_d y f son, respectivamente, los invariantes buscados, los invariantes de región y el valor de la función de costo a optimizar.

4.1 Comentarios

Comentario 1. La fase de exploración se realiza mientras la aptitud promedio de la población se mantenga en un valor muy alejado del valor óptimo.

Comentario 2. La fase de explotación se realiza cuando la aptitud promedio de la población se encuentra cerca del valor óptimo.

Comentario 3. La fase optimización local permite tomar una región generada por el AG, que posee un valor de aptitud muy alto, y procesarla para encontrar la región mínima que cumpla con la función de costo. Con esto se evita que el AG utilice generaciones adicionales para obtener la región reducida.

Comentario 4. Al usar múltiples puntos de cruce se logra manipular el cromosoma y el resultado siempre corresponde a regiones válidas dentro de la imagen.

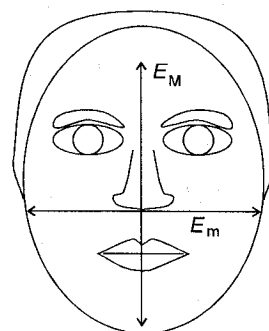


Figura 2. Ejes principales menor y mayor de un objeto, un rostro en este caso

¹ En el caso de imágenes pequeñas, el AG se comporta como un algoritmo de búsqueda local con la diferencia de que explora de manera simultánea diferentes regiones del espacio de representación. Para imágenes más grandes el número de generaciones requerido es mayor.

El AG inicia generando la población inicial que corresponde a regiones de tamaño igual a una celda. Cada cromosoma representa una región de tamaño igual a una celda. La siguiente población se obtiene mediante la cruce y/o mutación de los cromosomas de la población inicial. Esta población contendrá cromosomas correspondientes a regiones de con más de una celda. Este proceso se repite hasta obtenerse la mejor aproximación al mínimo de la función de costo.

Una descripción completa de como los algoritmos genéticos trabajan puede encontrarse en (Goldberg, 89).

5 Resultados Experimentales

En esta sección se prueba el desempeño de la técnica en dos escenarios diferentes. En primer lugar con objetos muy simples como caracteres impresos. En segundo lugar con objetos complicados como ojos y bocas humanas.

Algunas imágenes se obtuvieron con un escáner Logitech ScanMan 256 a 256 niveles de gris, otras se obtuvieron por medio de una con una cámara SONY FX-410 y una tarjeta digitalizadora Creative Labs, modelo Video Blaster SE100 a 24 bits de color. Estas imágenes se convirtieron después a 256 niveles de gris.

Todos los experimentos se realizaron en una computadora personal equipada de un procesador Pentium Intel a 200 Mhz y 32 Mbytes de memoria RAM.

5.1 Caso de Caracteres Impresos

5.1.1 Entrenamiento

El objeto a verificar en este experimento es uno de los siguientes: A, C, D, F y O. En cada caso nueve imágenes del objeto de interés en diferentes posiciones, orientaciones y tamaños fueron obtenidas. En cada imagen, el objeto fue segmentado manualmente. A partir de la región circundante, un vector de cuatro atributos fue calculado (Sección 2). A partir de los resultantes nueve vectores, un vector promedio fue obtenido. Este vector representa así el modelo del objeto a buscar, una letra en este caso.

5.1.2 Detección

Cien imágenes conteniendo los caracteres A, C, D, F y/o O, y algunas veces otros como E, P, R, V y X fueron usados para probar el sistema. Todas las imágenes fueron discretizadas a 200 X 200 pixeles y 256 niveles de gris. Una muestra de tales imágenes se muestra en la Figura 3.

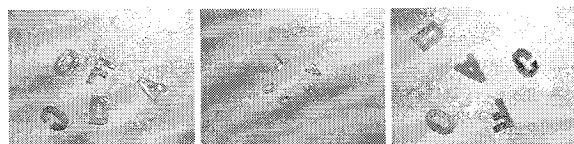


Figura 3. Una muestra de las imágenes usadas en la experimentación

El experimento consistió en realizar simulaciones tratando de encontrar una de las letras de interés (A, C, D, F y O). La búsqueda fue individual (una letra a la vez), realizando así 100 corridas por letra.

Durante la experimentación, con el AG se pudo segmentar satisfactoriamente en el 90% de los casos el objeto deseado. En el 20% de los casos, sin embargo, el AG generó falsos positivos. Las Figuras 4 (a) y (b) presentan algunos de los resultados. Los resultados numéricos se resumen en las Tablas 1 y 2.

Letra	Mayor diferencia	Relación de áreas	Mejor aprox.	Peor aprox.
A	10	1.10	0.9250	0.8995
C	7	1.20	0.9000	0.7000
D	12	0.98	1.0000	0.8300
F	9	0.95	0.9500	0.8900
O	10	1.08	0.9500	0.9000

Tabla 1. Resultados de simulación en términos de la eficacia de la segmentación con el AG

Letra	% Falsos Positivos	% Letra no encontrada	% Letra encontrada
A	22	8	92
C	16	10	90
D	19	11	89
F	18	9	91
O	25	12	88

Tabla 2. Resultados de la experimentación en términos de la eficiencia de la búsqueda

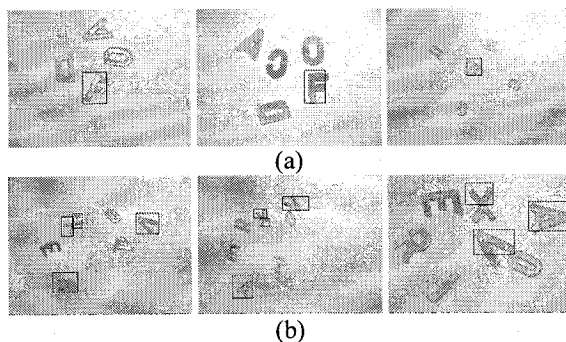


Figura 4. (a) Ejemplos de localización de la letra F. (b) Ejemplos de localización de la letra A. Nótese como en algunos Casos algunas letras son reconocidas por otras

En ambos casos, se obtuvieron los siguientes coeficientes:

- **Mayor diferencia.** Se refiere a la diferencia en pixeles entre los centroides de la región real (aquella segmentada manualmente) y la región segmentada a través del algoritmo genético. Este valor fue obtenido como promedio de los 100 casos.

- **Relación de áreas.** Este valor es la razón de áreas, entre la región real (aquella segmentada manualmente) y la región segmentada con el algoritmo genético. Un valor mayor a 1 indicaría que se obtuvo un área mayor que la real (no es un área minimal), mientras que un valor menor a 1 indicaría que se obtuvo un área menor que la real. Este es un valor promedio de los 100 casos.
- **Mejor aproximación.** Se refiere a la mejor aproximación de la función de aptitud, la cual dirige la búsqueda en el espacio de representación, obtenida en todo el proceso de simulación. Este es un valor promedio de los 100 casos.
- **Peor aproximación.** Se refiere a la peor aproximación de la función de aptitud, la cual dirige la búsqueda en el espacio de representación, obtenida en todo el proceso de simulación. Este es un valor promedio de los 100 casos.
- **% Falsos Positivos.** Porcentaje de casos en los que se segmentó objetos que no corresponden al objeto buscado.
- **% Letra no encontrada.** Porcentaje de casos en los que no se encontró el objeto de interés cuando sí existía en la imagen.
- **% Letra encontrada.** Porcentaje de casos en los que se encontró satisfactoriamente el objeto de interés.

5.2 Caso de Componentes Faciales Humanas

5.2.1 Entrenamiento

Para este segundo experimento, los objetos de interés fueron el ojo izquierdo, el ojo derecho y la boca de un ser humano. Estos objetos fueron escogidos ya que son muy buenas características para el reconocimiento de rostros. Para entrenar el sistema se usaron veinte imágenes de diferentes personas.

Cada imagen contiene un sólo rostro con variaciones en posición, rotación, cambios de escala y aún con cambios pequeños en expresión facial. Tres clases diferentes, una para el ojo izquierdo, una para el ojo derecho y otra para la boca fueron consideradas.

En el caso del objeto boca, por ejemplo, se obtuvo manualmente un rectángulo mínimo para cada una de las veinte imágenes de entrenamiento. A partir de cada rectángulo, los primeros cuatro invariantes de Hu fueron evaluados. De los veinte vectores resultantes, el vector promedio fue obtenido. Este vector representa el modelo deseado de la boca. Este mismo procedimiento fue aplicado para el ojo izquierdo y para el derecho.

5.2.2 Detección y Localización

Doscientas cuarenta imágenes conteniendo al menos un rostro y diferentes fondos fueron usados para probar el desempeño de la técnica propuesta. En este caso, todas las imágenes fueron discretizadas a 200 X 200 y 16 niveles de gris con cambios ligeros en posición, rotación y tamaño y aún pequeños en la expresión en los rostros.

Durante las pruebas, el AG encontró satisfactoriamente en el 100% de los casos el rasgo facial buscado. En el 20% de los

casos, sin embargo, genera falsos positivos. En la Figura 5 se muestran algunos ejemplos obtenidos durante la experimentación. Los resultados numéricos son resumidos en las Tablas 3 y 4.



Figura 5. Algunos ejemplos de los resultados obtenidos durante la experimentación

Rasgo Facial	* Mayor diferencia	* Mejor aproximación	Peor aproximación
1	7	0.931	0.85
2	5	0.907	0.82
3	8	0.924	0.78

Tabla 4. Resultados experimentales en términos de la eficiencia del AG para detectar y localizar rasgos faciales

Rasgo Facial	% de rasgos faciales encontrados	% de falsos positivos
1	100	7
2	100	4
3	100	9

Tabla 5. Resultados experimentales en términos de la eficiencia para la búsqueda del AG

En ambos casos:

- **% de rasgos faciales encontrados.** Porcentaje de casos en los que se encontró satisfactoriamente el rostro de deseado.
- **% Falsos Positivos.** Porcentaje de casos en que los rostros segmentados no corresponden al deseado.
- ***** Representa el valor promedio de los 240 casos.

5.2.3 Tiempos de Ejecución

Como se indicó en la Sección 4, un AG permite la evolución de los organismos más aptos dentro de un entorno. Esta evolución se logra con la acumulación de características, de una generación a otra. En términos reales, se requiere de una gran cantidad de generaciones para que una especie evolucione y se adapte a su entorno.

Es bien sabido que un AG requiere de grandes cantidades de procesamiento; sin embargo, para ciertas aplicaciones es deseable que el AG responda en tiempos razonables (segundos, a lo más horas) con una buena aproximación de la solución del problema. En nuestro caso vimos con 5 poblaciones el sistema convergía a una solución. Tomando en cuenta que por cada población se requiere de alrededor de

medio segundo, el tiempo total de procesamiento fue de aproximadamente 2.5 segundos.

6 Conclusiones

En este trabajo se presentó una técnica útil para la detección y la localización automática de objetos a partir de una imagen 2D en niveles de gris. La técnica se vale de un algoritmo genético y una descripción invariante del objeto para llevar a cabo la tarea especificada. Los descriptores usados son los primeros cuatro invariantes a traslaciones, rotaciones y cambios de escala propuestos por Hu.

En una primera etapa, una imagen posiblemente conteniendo un objeto de interés es dividida en pequeñas celdas rectangulares cuyo tamaño depende de la aplicación. Para cada celda, los momentos ordinarios son calculados. A partir de estas cantidades, los correspondientes invariantes de Hu son derivados. La detección de objetos es llevada a cabo por el agrupamiento de celdas individuales usando un algoritmo genético a través del ajuste de una función de costo específica. Esta función involucra la descripción del objeto.

El uso de la técnica propuesta evita la realización del proceso exhaustivo de búsqueda. En este caso, para cada punto en la imagen, ventanas de diferentes tamaños tienen que ser evaluadas, y para cada una de estas ventanas, el vector de descriptores tiene que ser calculado y comparado con el vector modelo. El AG barre la imagen completa considerando únicamente regiones de interés guiadas por la función de costo. El tiempo promedio para localizar un objeto en una computadora personal a 200 Mhz es de menos de tres segundos.

6.1 Ventajas de la Metodología

La metodología permite utilizar cualquier función que pueda ser calculada directamente sobre una imagen, como elemento descriptor de los objetos (por ejemplo: área, perímetro, entropía, momentos, color, etcétera).

Otra ventaja está asociada al paralelismo natural de los AG's. Un AG puede procesar y evaluar de manera simultánea diferentes elementos de la población (posibles soluciones) y recorrer, dentro del espacio de representación o espacio de soluciones (ver Anexo A), diferentes regiones de manera simultánea. Esto permite superar las deficiencias que presentan algunos métodos de búsqueda (por ejemplo el de escalada de la colina) que convergen prematuramente a mínimos o máximos locales. Este paralelismo es una característica muy interesante desde el punto de vista del procesamiento computacional ya que un AG se puede implementar en una computadora, o en varias conectadas en red local o remota (Internet), y el proceso o la carga se distribuye con lo que los tiempos se reducen.

6.2 Desventajas de la Metodología

Un inconveniente de la técnica propuesta es que existen zonas dentro de la imagen que generan valores similares para los descriptores y ocasiona que se generen falsos positivos que se presentan como regiones que no contienen al objeto de interés pero que el AG, con la ayuda de los descriptores invariantes,

considera que sí. Normalmente, esto se corrige incorporando una etapa de validación y eliminación de falsos positivos al terminar la ejecución del AG.

En este trabajo se reporta el uso de los invariantes de Hu como descriptores. Como se sabe estos descriptores se comportan muy bien (son robustos) ante cambios de tamaño, orientación y traslación pero no a cambios de contraste (cambios en la iluminación de la imagen). Para lograr que los descriptores sean poco sensibles a los cambios de iluminación se puede realizar un procesamiento previo sobre la imagen; se pueden utilizar descriptores como los momentos invariantes de Maítra (Maítra, 79) que son robustos a cambios de contraste, rotación, escala y traslación.

Otra desventaja está asociada a la naturaleza del AG: en general el número de generaciones requeridas para resolver un problema es muy grande con los correspondientes recursos de cómputo (tiempo de proceso, espacio de trabajo, etcétera).

Los AG's se han evaluado y comparado con otros métodos de optimización y en general se comportan mejor que la mayoría de ellos. Un AG, al igual que las otras técnicas de búsqueda y heurísticas, no garantiza que se encuentre el óptimo.

Agradecimientos

Este trabajo fue realizado con apoyo económico del Consejo Nacional de Ciencia y Tecnología (CONACyT) a través de su programa de becas y mediante el financiamiento de proyectos con la referencia 28949-A.

Anexo A. Cardinalidad de E

En el peor caso, la cardinalidad c de E puede aproximarse por la suma de los números de Stirling (Graham, 1992) la cual corresponde al número de formas diferentes en que un conjunto de datos puede particionarse, en este caso el conjunto de píxeles que forman la imagen. De acuerdo a (Graham, 92), la cardinalidad c de E puede obtenerse como:

$$c = \sum_{k=0}^K \left\{ \begin{matrix} n \\ k \end{matrix} \right\} \quad (\text{A.1})$$

En esta caso k es el tamaño de las particiones, K es el tamaño máximo para las particiones y n es el número de objetos que se desea particionar. Los números de Stirling pueden obtenerse por medio de la función de recurrencia:

$$\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = k \left\{ \begin{matrix} n-1 \\ k \end{matrix} \right\} + \left\{ \begin{matrix} n-1 \\ k-1 \end{matrix} \right\}, \quad n \in \mathbb{Z}^+$$

$$\left\{ \begin{matrix} 0 \\ 0 \end{matrix} \right\} = 1, \quad \left\{ \begin{matrix} n \\ 0 \end{matrix} \right\} = 0$$

Por ejemplo, si $n=3$:

$$\sum_{k=0}^3 \binom{3}{k} = \binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 0 + 1 + 3 + 1 = 5.$$

Para tener una idea del comportamiento de *c*, en la Tabla A se presenta el número de particiones con cardinalidad fija y el total de particiones para tamaños de población de 0 a 7.

De acuerdo con esta tabla, existen 877 formas diferentes de particionar 7 píxeles. Afortunadamente, no todas las configuraciones dentro de las particiones son válidas por no cumplir con ciertas características como: conexidad, forma de la región, área mínima, área máxima, etc. el número se reduce.

En el mejor caso se puede considerar que en cada punto de la imagen se evalúa una región arbitraria de dimensiones variables. El número total de ventanas o regiones a ser evaluadas para una imagen se puede aproximar por:

$$c = ren \times col \times ven \tag{A.2}$$

En este caso, *c* es el número total de regiones que se evalúan, *ren* es el número de renglones de la imagen, *col* es el número de columnas de la imagen, *ven* es el número de ventanas que se evalúan en cada pixel y *ren x col* es el número de píxeles de la imagen.

Para tener una idea de que tan grande es *c* considérese el siguiente ejemplo. Para una imagen de 100 x 100 píxeles y considerando ventanas cuadradas con tamaños que varían desde 20 X 20 hasta 60 X 60 píxeles y con incrementos de 2 píxeles por lado, el número total de ventanas en un punto cualquiera de la imagen sería de 20. Si suponemos que sólo se evalúan ventanas incluidas completamente en la imagen, entonces el número total de regiones (*c*) sería 78680.

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\sum_{i=0}^n \binom{n}{i}$
0	1								
1	0	1							1
2	0	1	1						2
3	0	1	3	1					5
4	0	1	7	6	1				15
5	0	1	15	25	10	1			52
6	0	1	31	90	65	15	1		203
7	0	1	63	301	350	140	21	1	877

Tabla A. El número de formas diferentes en que se pueden particionar *n* datos viene dado por el valor de la última columna de la derecha. Para *n*=7 existen 877 particiones diferentes

Referencias

Hu, M. K., Visual Pattern recognition by moment invariants, IRE Transactions on Information Theory, pp. 179-187, 1962.

Guzmán, A., Computer recognition of three-dimensional objects in a visual scene, Reporte de Tesis doctoral, MAC-TR-59, MIT, December, 1968.

Dudani, S. A. et al. Aircraft identification by moment invariants, IEEE Transactions on Computers, 26(1):39-46, 1977.

Perkins, W. A., A model-based vision system for industrial parts, IEEE Transactions on Computers, 27:123-143, 1978.

Davis, L. S., Shape matching using relaxation techniques, IEEE Transactions on Patt. Anal. and Mach. Intell. 1(1):60-72, 1979.

Bhanu, B. and Faugeras, O., Shape matching of two-dimensional objects, IEEE Transactions on Patt. Anal. and Mach. Intell. 6:137-156, 1984.

Ayache, N. and Faugeras, O., HYPER-A new approach for the recognition and position of two-dimensional objects, IEEE Transactions on Patt. Anal. and Mach. Intell.: 44-54, 1986.

Sossa, H. and Horaud, R., Model Indexing: The Graph-Hashing Approach. IEEE Conference on Computer Vision and Pattern Recognition, Champaign, Illinois, USA. 15--18 June, pp. 811--814, 1992.

Reiss, T. R., Object recognition using algebraic and differential invariants, Signal Processing, 32:367-395, 1993.

Jacobsen, S. B. and Jensen, M. S., Model-based recognition of planar objects under perspective views using geometric invariants, Proceedings of the SCIA. pp. 411-418, 1993.

Rothwell, C. A., Zisserman, A., Forsyth, D. A. and Mundy, J. L., Planar object recognition using projective shape representations. International Journal of Computer Vision, 16:57-99, 1995.

Sossa, H. y Palomino, A., Reconocimiento de objetos 2-D parcialmente ocultos por indexado a través de rasgos visuales locales, Soluciones Avanzadas, número especial sobre Visión por Computadora, No. 42. pp. 23-31, 1997.

Sossa, H., Rayón, P. and Figueroa-Nazuno, J., 2D object recognition by indexing through a modified ART2 Neural Network, Expert Systems with Applications, 14(1):199-210, 1998.

González, R. C. y Woods, R. E., Tratamiento Digital de Imágenes, Addison-Wesley/Diaz de Santos, 1996.

Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley Publishing Company, inc. 1989.

Pal, S. K. and Wang, P. P., Genetic algorithms for pattern recognition, CRC Press, Inc. 1996.

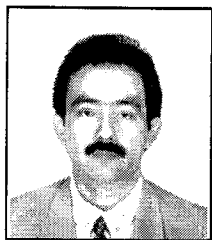
Davidor, Y., Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization. World Scientific Publishing Co. 1991.

Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag Berlin Heidelberg, New York, 1992.

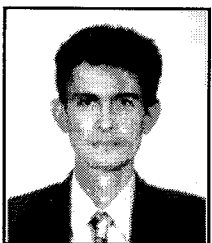
Holland, J., Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michigan Press. 1975.

Maître, S., Moment Invariants, Proc. IEEE 67, 697-699, 1979.

Graham, R. L., Knuth, D. E. and Patashnik, O., Concrete mathematics: a foundation for computer science, Addison Wesley Publishing Company, 1992, páginas 243-253.



***Raúl Pinto Elías**, (1965) recibió el grado de Maestro en Ciencias en Ciencias Computacionales por el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) en 1993 y el grado de Doctor en Ciencias en Ingeniería Eléctrica del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN) en el 2000. Es Profesor Investigador Titular en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET). Sus áreas de interés incluyen la visión por computadora, el reconocimiento de patrones, la visión robótica y sus aplicaciones a procesos de inspección y manufactura.*



***Juan Humberto Sossa Azuela**, (1956) recibió el grado de Maestro en Ciencias en Ingeniería Eléctrica del Centro de Investigación y Estudios Avanzados del IPN de México (CINVESTAV-IPN), en 1987 y el grado de Doctor en Informática del Instituto Nacional Politécnico de Grenoble, Francia en 1992. Es profesor Titular en el Centro de Investigación en Computación del IPN. Sus áreas de interés incluyen la visión por computadora, el procesamiento de imágenes y el control de sistemas electromecánicos usando información visual.*

