



El algoritmo “Artificial Bee Colony” (ABC) y su uso en el Procesamiento digital de Imágenes

Erik Cuevas

CUCEI, Universidad de Guadalajara

Av. Revolución 1500, C.P. 44430, Guadalajara, Jalisco, México.

erik.cuevas@cucei.udg.mx

Resumen Métodos tradicionales de procesamiento de imagen presentan diferentes dificultades al momento de ser usados en imágenes que poseen ruido considerable y distorsiones. Bajo tales condiciones, el uso de técnicas de optimización bio-inspiradas se ha extendido en los últimos años. Este artículo explora el uso del algoritmo de optimización “Artificial Bee Colony” (ABC) para el procesamiento digital de imágenes. ABC es un algoritmo metaheurístico inspirado en el comportamiento colectivo de las abejas, el cual ha sido empleado exitosamente para resolver problemas complejos de optimización. En este artículo se presentan dos problemas representativos en el área de procesamiento digital de imágenes, como lo son: la segmentación y la detección de círculos, los cuales son planteados desde el punto de vista de optimización. Considerando este enfoque, en la segmentación, el histograma de una imagen es aproximado por medio de la mezcla de funciones gaussianas que representan las clases contenidas en la imagen y cuyos parámetros son calculados por el algoritmo ABC. Por otro lado, en la detección de círculos se utiliza una combinación de tres puntos borde para codificar círculos candidatos. Utilizando las evaluaciones de una función objetivo (que determina si tales círculos están presentes en la imagen) el algoritmo ABC realiza una exploración eficiente hasta encontrar el círculo que mejor se aproxime a aquel contenido en la imagen. Resultados experimentales muestran que las soluciones generadas son capaces de resolver adecuadamente los problemas planteados.

Abstract Classical methods often face great difficulties in solving image processing problems in images containing noise and distortions. Under such conditions, the use of bio-inspired optimization approaches has been extended. This paper explores the use of the Artificial Bee Colony (ABC) algorithm for digital image processing seen as an optimization problem. ABC is a heuristic algorithm motivated by the biological behaviour of honeybees which has been successfully employed to solve complex optimization problems. In this paper, image segmentation and circle detection tasks are considered as examples, both issues approached as optimization problems. In segmentation, an image 1-D histogram is approximated through a Gaussian mixture model whose parameters are calculated by the ABC algorithm. On the other hand, the circle detector uses a combination of three edge points as parameters to construct candidate circles. A matching function determines if such candidate circles are actually present in the image. Experimental results show that the generated solutions are able to solve properly the considered problems.

Palabras clave: Algoritmo Artificial Bee Colony, algoritmos metaheurísticos, procesamiento inteligente de imágenes, segmentación de imágenes, detección de círculos.

Keywords: Artificial Bee Colony algorithm, nature inspired algorithms, intelligent image processing, image segmentation, circle detection.

1 Introducción

Durante la última década se ha presentado un crecimiento sostenido en el campo de los algoritmos bio-inspirados de cómputo evolutivo para la búsqueda y optimización. Algunos ejemplos de tales enfoques son: los algoritmos genéticos (GA,[1]) y evolución diferencial (DE, [2]), los cuales han sido aplicados para resolver problemas complejos de optimización. Una gran parte de los algoritmos bio-inspirados consideran el fenómeno de inteligencia en enjambre (“swarm intelligence”, [3]) como fuente de inspiración. Tal enfoque sugiere, que agentes con operaciones muy sencillas permiten resolver problemas complejos mediante la iteración de sus comportamientos. Clases especiales de tales enfoques son los algoritmos creados por analogía de parvadas de aves; como lo es el algoritmo de optimización de enjambre de partículas (PSO, [4]), por el comportamiento de colonias de hormigas; como lo es el algoritmo de optimización de colonia de hormigas (ACO, [5]) y por la manera en la que se alimentan las bacterias; como lo es el algoritmo optimización de búsqueda de alimento de bacterias (BFOA, [6]).

Karaboga presentó en 2005 un algoritmo de enjambre de abejas para resolver problemas numéricos de optimización [7], el cual es conocido como el método “artificial bee colony” (ABC). Inspirado por el comportamiento biológico de las colonias de abejas en su búsqueda por alimento, el algoritmo ABC consiste de tres componentes esenciales: posiciones de fuentes de alimento, cantidades de néctar y diferentes clases de abejas. En el algoritmo, cada posición de fuente de alimento representa una solución plausible del problema bajo consideración y la cantidad de néctar representa la calidad de dicha solución. Cada clase de abeja representa una operación particular para generar nuevas posiciones de fuentes de alimento (nuevas soluciones).

Por otro lado, una de las operaciones más importantes en el procesamiento digital de imágenes es la segmentación, la cual busca agrupar píxeles con características comunes en regiones significativas. El establecimiento de un umbral ha sido usado como una herramienta simple pero eficiente para segmentar objetos de interés. En la literatura una gran variedad de enfoques basados en umbral para dos clases han sido propuestas para la segmentación de imágenes, tales métodos incluyen algoritmos clásicos [8-11] y técnicas inteligentes [12-14]. Sin embargo, extender las capacidades de tales algoritmos para la segmentación de múltiples clases puede causar algunos inconvenientes tales como: (1) no contar con una solución analítica, (2) baja convergencia o (3) alto costo computacional.

Las mezclas Gaussianas representan un método flexible de modelado estadístico, el cual es empleado en una gran variedad de contextos [15]. Este modelo ha recibido una atención considerable en el desarrollo de algoritmos de segmentación. Varios algoritmos han sido propuestos para resolver el problema de aproximación de mezclas Gaussianas siendo el Expectación-Maximización (EM) [16] y el Levenberg-Marquadt (LM) [17] dos de los más empleados. A pesar de que estos métodos presentan en general un buen desempeño, existen condiciones bajo las cuales presentan fuertes inconsistencias, tales como la sensibilidad a sus condiciones iniciales y falta de precisión en los parámetros estimados [18].

En este trabajo se presenta un enfoque alternativo de segmentación basado en el algoritmo de optimización ABC. En dicho método, el histograma de una imagen es aproximado a través de la mezcla de un conjunto de funciones gaussianas, las cuales representan a cada una de las clases (umbrales). Usando una función objetivo que mide la semejanza entre la mezcla y el histograma, el algoritmo ABC modifica los parámetros de cada una de las funciones, hasta que la aproximación sea la mejor posible.

El problema de detectar primitivas circulares también tiene gran importancia dentro del análisis de imágenes en aplicaciones industriales tales como la inspección automática de productos manufacturados, el reconocimiento de objetivos, etc. [19]. La detección de círculos en imágenes digitales es comúnmente resuelta a través de la transformada circular de Hough [20]. Desafortunadamente este enfoque requiere una gran cantidad de espacio en memoria, lo que produce una baja velocidad de cómputo [21]. Para poder superar este problema algunos investigadores han propuesto nuevos enfoques siguiendo los principios de la transformada de Hough, dando como resultado la transformada probabilística de Hough [22], la transformada aleatoria de Hough (RHT, [23]), la transformada difusa de Hough [24] y algunas otras discutidas en [25].

Como una alternativa a las técnicas basadas en la transformada de Hough, el problema de detección de formas ha sido tratado por medio de métodos basados en cómputo evolutivo. En general estos algoritmos han demostrado presentar mejores resultados que aquellos basados en la transformada de Hough en cuanto a precisión, velocidad y

robustez [26]. En estos métodos el proceso de detección se traduce a un problema de optimización el cual envuelve el uso de una función objetivo que evalúa la calidad de la detección. Como resultado, varios detectores de círculos, basados en estos enfoques, han sido propuestos considerando diferentes técnicas de computo evolutivo tales como: Algoritmos genéticos (GA) [26], optimización de búsqueda de alimento de bacterias (BFOA) [27], búsqueda armónica (HS) [28] y evolución diferencial (DE) [29].

En este artículo se presenta un algoritmo para la detección automática de múltiples formas circulares en imágenes ruidosas y complicadas, el cual no toma en consideración los principios convencionales de la transformada de Hough. Considerando este enfoque se utiliza una combinación de tres puntos borde para codificar los círculos candidatos (posibles soluciones), mientras que una función objetivo es utilizada para medir la existencia de un círculo candidato sobre la imagen de bordes. Guiado por los valores de tal función objetivo, el conjunto de círculos candidatos es modificado por medio del algoritmo ABC de manera tal que el mejor círculo candidato pueda ser ajustado dentro de la figura más circular contenida en la imagen de bordes.

El artículo está organizado de la siguiente manera: La sección 2 describe el funcionamiento y características del algoritmo ABC, mientras que la sección 3 detalla el procedimiento para segmentación de imágenes digitales considerando el algoritmo ABC. En la sección 4 se describe el enfoque del detector de círculos y finalmente la sección 5 presenta las conclusiones obtenidas.

2 Algoritmo artificial bee colony (ABC)

El algoritmo ABC define un conjunto de operaciones que asemejan algunas características del comportamiento de las abejas. Cada solución dentro del espacio de búsqueda incluye un conjunto de parámetros que representan las posiciones de las fuentes de alimento. El valor de “afinidad” (proporcionado por la función objetivo) hace referencia a la calidad de la fuente de alimento. En general, el proceso de optimización imita la búsqueda de las abejas por fuentes importantes de alimento dando como resultado un proceso análogo a encontrar soluciones óptimas.

2.1 Descripción del algoritmo ABC

Semejante a otros enfoques metaheurísticos, el algoritmo ABC realiza un proceso iterativo el cual es repetido durante un determinado número NG de iteraciones. Este comienza con una población de soluciones o fuentes de alimento generadas aleatoriamente, mientras que las siguientes tres operaciones son aplicadas hasta que un criterio de paro es alcanzado [30]:

1. Enviar abejas obreras.
2. Seleccionar fuentes de alimento por parte de abejas espectadoras.
3. Determinar las abejas que serán exploradoras.

2.2 Inicialización de la población

El algoritmo comienza inicializando N_p fuentes de alimento para las abejas obreras; cada fuente de alimento simboliza un vector de D elementos que representa las variables de decisión, las cuales son aleatoriamente determinados entre los límites inferiores x_j^{low} y superiores x_j^{high} previamente definidos.

$$x_{j,i} = x_j^{low} + \text{rand}(0,1) \cdot (x_j^{high} - x_j^{low}); \quad j = 1, 2, \dots, D; \quad i = 1, 2, \dots, N_p \quad (1)$$

siendo j e i los índices del parámetro y población respectivamente. Por lo tanto $x_{j,i}$ es el j -ésimo parámetro del i -ésimo individuo.

2.3 Enviar abejas obreras

En esta operación cada abeja obrera genera una nueva fuente de alimento en la vecindad de su posición actual de la siguiente manera:

$$v_{j,i} = x_{j,i} + \phi_{j,i}(x_{j,i} - x_{j,k}); k \in \{1, 2, \dots, N_p\}; j \in \{1, 2, \dots, D\} \quad (2)$$

donde $x_{j,i}$ es un parámetro j seleccionado aleatoriamente del i -ésimo individuo y k es una de las N_p fuentes de alimento, satisfaciendo la condición $i \neq k$. Si un parámetro dado de la solución candidata v_i excede sus límites predeterminados, ese parámetro debe ser ajustado de manera tal que se encuentre en el rango definido. El factor de escalamiento $\phi_{j,i}$ es un número aleatorio entre $[-1, 1]$. Una vez que una nueva solución ha sido generada, se calcula su calidad mediante una función objetivo. La calidad fit_i de una solución candidata v_i en el contexto de ABC para un problema de minimización se asigna a través de la siguiente expresión:

$$fit_i = \begin{cases} \frac{1}{1 + J(v_i)} & \text{if } J(v_i) \geq 0 \\ 1 + |J(v_i)| & \text{if } J(v_i) < 0 \end{cases} \quad (3)$$

$J(\cdot)$ es la función objetivo a ser minimizada. Si la cantidad de néctar (calidad de la solución) de v_i es mayor, entonces la solución x_i es reemplazada por v_i ; en otro caso x_i permanece.

2.4 Selección de fuentes de alimento por abejas espectadoras

Cada abeja espectadora selecciona una de las fuentes de alimento propuestas dependiendo de su calidad. La probabilidad de que una fuente de alimento sea seleccionada se obtiene a partir de la siguiente ecuación:

$$Prob_i = \frac{fit_i}{\sum_{i=1}^{N_p} fit_i} \quad (4)$$

donde fit_i es el valor de calidad de la fuente de alimento i . La probabilidad de que una fuente de alimento sea seleccionada por una abeja espectadora incrementa con un aumento en el valor de calidad de la fuente de alimento. Después de que la fuente de alimento es escogida, las abejas espectadoras irán a la posición seleccionada y determinarán una nueva fuente de alimento candidata dentro de la vecindad de la fuente seleccionada. Tal posición se calcula por medio de (2). En caso de que la calidad de la nueva solución, sea mejor que antes, dicha posición es mantenida; en otro caso la última solución se reemplaza.

2.5 Determinar abejas exploradoras

Si una fuente de alimento i (solución candidata) no puede ser mejorada a lo largo de un número predeterminado de L intentos. La fuente de alimento se abandona y la abeja correspondiente se convierte en una exploradora. Para verificar si una solución candidata ha alcanzado el límite L predeterminado, un contador A_i es asignado a cada fuente de alimento i . Dicho contador es incrementado como consecuencia de que una operación de abeja falle en mejorar la calidad de una solución.

3 Segmentación multi-umbral

Se considera una imagen con L niveles de gris $[0, \dots, L-1]$ cuya distribución es representada por el histograma $h(g)$. Para simplificar la descripción, el histograma es normalizado tal como una función de distribución de probabilidad, dando como resultado:

$$h(g) = \frac{n_g}{N}, \quad h(g) > 0, \quad N = \sum_{g=0}^{L-1} n_g, \quad \text{y} \quad \sum_{g=0}^{L-1} h(g) = 1, \quad (5)$$

donde n_g denota el número de píxeles con nivel de gris g y N el número total de píxeles en la imagen. El histograma puede por lo tanto ser aproximado por una mezcla de funciones gaussianas definida de la forma:

$$p(x) = \sum_{i=1}^K P_i \cdot p_i(x) = \sum_{i=1}^K \frac{P_i}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right] \quad (6)$$

siendo P_i la probabilidad a priori, μ_i la media y σ_i la desviación estándar de la i -ésima función de distribución de probabilidad. K representa el número total de clases de la imagen. Adicionalmente, la restricción $\sum_{i=1}^K P_i = 1$ debe satisfacerse. La calidad de la aproximación entre la mezcla de gaussianas y el histograma es medida mediante la siguiente función objetivo:

$$J = \frac{1}{n} \sum_{j=0}^{L-1} [p(x_j) - h(x_j)]^2 + \omega \cdot \left| \left(\sum_{i=1}^K P_i \right) - 1 \right|, \quad (7)$$

asumiendo un histograma de L niveles de gris y ω es la penalidad asociada con la restricción. En general, la estimación de los parámetros que minimizan la función objetivo (7) no es un problema simple. El uso de métodos de optimización clásicos tales como el Expectación-Maximización [25] o los basados en gradiente [26] pueden quedar atorados dentro de mínimos locales (sensibles a condiciones iniciales) y ser costosos en tiempo.

3.1 Determinación de los valores de umbral

Finalmente para determinar los valores de umbral óptimos se considera que las clases están organizadas de manera tal que $\mu_1 < \mu_2 < \dots < \mu_K$. Por lo tanto, los valores de umbral son obtenidos calculando la probabilidad de error total de dos funciones Gaussianas adyacentes, resultando:

$$E(T_h) = P_{h+1} \cdot E_1(T_h) + P_h \cdot E_2(T_h), \quad h = 1, 2, \dots, K-1 \quad (8)$$

tal que:

$$E_1(T_h) = \int_{-\infty}^{T_h} p_{h+1}(x) dx \quad \text{y} \quad E_2(T_h) = \int_{T_h}^{\infty} p_h(x) dx, \quad (9)$$

donde $E_1(T_h)$ es la probabilidad de clasificar erróneamente los píxeles en la clase $(h+1)$, cuando realmente pertenecen a la clase h , mientras que $E_2(T_h)$ representa la probabilidad de clasificar inadecuadamente los píxeles en la clase h , cuando realmente corresponden a la clase $(h+1)$. Las P_j 's son las probabilidades a priori de la mezcla, y T_h es el valor de umbral de la h -ésima clase. El valor de T_h es seleccionado de manera tal que el error $E(T_h)$ sea minimizado. Diferenciando $E(T_h)$ con respecto a T_h e igualando el resultado a cero, es posible usar la siguiente ecuación para definir el valor óptimo de umbral T_h :

$$AT_h^2 + BT_h + C = 0, \quad (10)$$

considerando

$$\begin{aligned} A &= \sigma_h^2 - \sigma_{h+1}^2, \quad B = 2 \cdot (\mu_h \sigma_{h+1}^2 - \mu_{h+1} \sigma_h^2) \\ C &= (\sigma_h \mu_{h+1})^2 - (\sigma_{h+1} \mu_h)^2 + 2 \cdot (\sigma_h \sigma_{h+1})^2 \cdot \ln \left(\frac{\sigma_{h+1} P_h}{\sigma_h P_{h+1}} \right) \end{aligned} \quad (11)$$

A pesar de que (10) tiene dos posibles soluciones, solo una de ellas es plausible, *i.e.* un valor positivo que cae dentro del intervalo.

3.2 Resultados de la segmentación

Esta sección presenta dos experimentos que analizan el desempeño del algoritmo ABC para la segmentación de imágenes mediante la aproximación del histograma.

En el primer caso se considera la imagen conocida como “The Camera-man”, la cual es mostrada en la Figura 1a, con su correspondiente histograma en la Figura 1b. El objetivo es segmentar la imagen en tres diferentes clases. De esta manera el algoritmo ABC ajusta nueve diferentes parámetros, siguiendo el procedimiento de minimización dictado por la función objetivo definida en (7). Para este caso ABC fue configurado con una población de 40 abejas es considerada, con 20 (N_p) abejas obreras y 20 (N_p) espectadoras. Cada solución candidata es codificada de la siguiente manera:

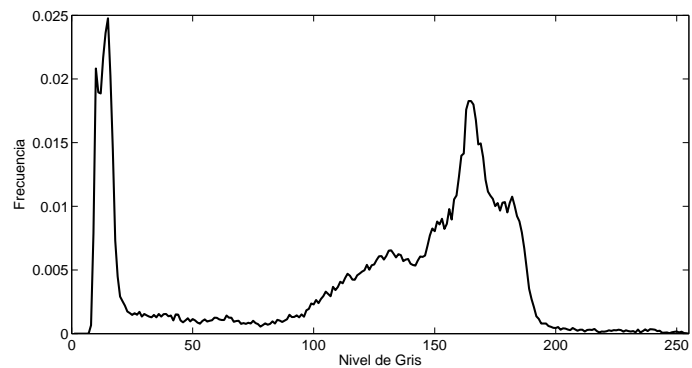
$$I_N = \{P_1^N, \sigma_1^N, \mu_1^N, P_2^N, \sigma_2^N, \mu_2^N, P_3^N, \sigma_3^N, \mu_3^N\} \quad (12)$$

donde N representa el número de individuo. Los parámetros ($P_i^N, \sigma_i^N, \mu_i^N$) son inicializados aleatoriamente pero asumiendo algunas restricciones para cada parámetro (por ejemplo μ_i^N es un valor entero entre 0 y 255).

Después de 200 iteraciones ($NG=200$) el algoritmo ABC ha convergido al mínimo global. La Figura 2a muestra las funciones gaussianas obtenidas (clases de píxel) graficadas sobre el histograma original, mientras que la Figura 2b muestra la mezcla gaussiana final. La Figura 3 muestra la imagen segmentada cuyos valores de umbral son calculados de acuerdo a las ecuaciones (10) y (11).



(a)



(b)

Figura 1. (a) imagen original “The Cameraman”, y (b) su histograma correspondiente.

En el segundo caso se considera la imagen de prueba “The scene” (véase Figura 4a). El histograma de la imagen es presentado en la Figura 4b. Similar al primer caso, la imagen es segmentada considerando ahora cuatro diferentes clases. La optimización es realizada por el algoritmo ABC cuyo resultado es mostrado en la Figura 5a. A su vez, la Figura 5b presenta la aproximación final como resultado de la adición de las funciones gaussianas. La Figura 6 muestra la segmentación de la imagen considerando el resultado obtenido.

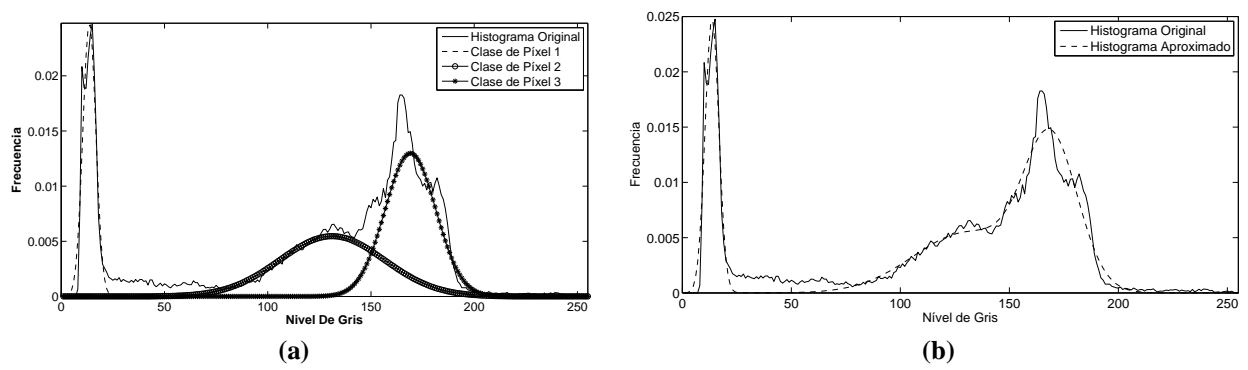


Figura 2. Aplicación del algoritmo ABC para 3 clases y sus resultados: (a) funciones gaussianas de cada clase y (b) aproximación final.



Figure 3. Imagen segmentada considerando solo tres clases.

3.3 Comparación del rendimiento de ABC

En esta sección se compara el desempeño de la propuesta ABC contra otros enfoques tales como el Expectación-Maximización [16] y el Levenberg-Marquadt (uno de los mejores algoritmos de optimización basado en gradiente, [17]), los cuales son comúnmente usados para determinar los parámetros de mezclas gaussianas. Con el objetivo de conducir los experimentos, una imagen sintética fue diseñada para ser usada como referencia en las comparaciones. El propósito es conocer con precisión el numero de clases y los parámetros que constituyen a la mezcla gaussiana que aproxima al histograma de la imagen. Tal imagen sintética es dividida en tres secciones. Cada sección corresponde a una clase diferente, la cual es producida configurando cada pixel v_i a un nivel de gris generado por la siguiente expresión:

$$v_i = e^{-\left(\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)}, \tag{13}$$

donde μ_i y σ_i representan la media y la dispersión de los niveles de gris de los pixeles de la sección i , respectivamente. La imagen sintética y su correspondiente histograma empleados en este estudio comparativo se muestra en la Figura 4. Ellos fueron generados considerando los parámetros mostrados en la Tabla 1.

La comparación entre los métodos se concentra en dos aspectos: la sensibilidad a las condiciones iniciales y el costo computacional.

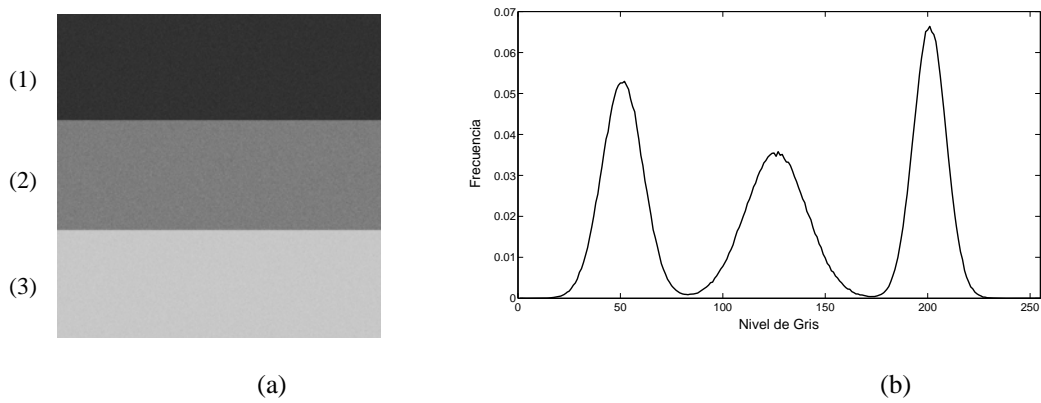


Figura 4. (a) Imagen sintética y (b) histograma correspondiente

Tabla 1 : Parámetros empleados para la generación de la imagen sintética de la Figura 3.

	P_i	μ_i	σ_i
(1)	0.053	50	10
(2)	0.035	125	15
(3)	0.067	200	8

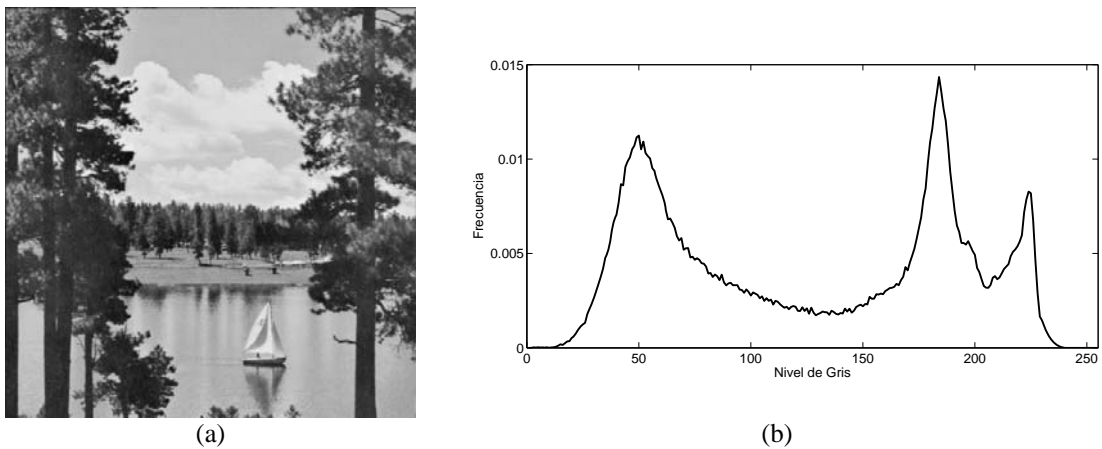


Figura 5. Segundo experimento, (a) imagen original "The scene", y (b) su histograma.

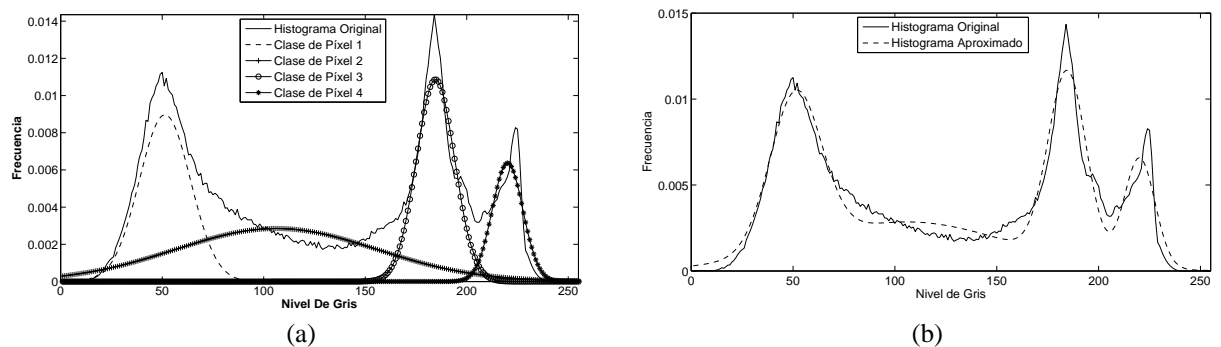


Figura 6. Resultados obtenidos por el algoritmo ABC para 4 clases: (a) funciones gaussianas para cada clase, (b) mezcla de funciones gaussianas aproximando el histograma original.



Figura 7. Imagen segmentada considerando cuatro clases.

a) *Sensibilidad a las condiciones iniciales.*

Todos los métodos empleados para aproximar mezcla de Gaussianas inician con una o varias soluciones iniciales. Después, considerando las características propias de cada algoritmo esta solución o soluciones son modificadas hasta alcanzar la mejor aproximación (solución) posible. En este experimento es evaluado la sensibilidad de los algoritmos cuando los parámetros de la mezcla candidata inicial, son configurados usando diferentes valores.

Tabla 2 : Comparación entre los algoritmos EM, LM y ABC, considerando diferentes valores iniciales.

	Valor Inicial (1)	E			Valor inicial (2)	E		
		EM	LM	ABC		EM	LM	ABC
μ_1	10	6.15	5.22	1.21	60	22.78	14.21	1.14
μ_2	81	9.41	8.04	2.54	150	30.14	20.07	1.87
μ_3	185	8.57	7.25	2.14	220	27.24	18.54	1.74
σ_1	8	0.3	0.24	0.14	15	4.51	3.11	0.16
σ_2	10	0.21	0.25	0.17	20	5.12	2.45	0.11
σ_3	6	0.14	0.10	0.05	10	3.9	1.78	0.08
P_1	0.025	1×10^{-3}	0.88×10^{-3}	0.11×10^{-3}	0.060	3×10^{-3}	2.7×10^{-3}	0.10×10^{-3}
P_2	0.025	0.9×10^{-3}	0.54×10^{-3}	0.08×10^{-3}	0.040	4.1×10^{-3}	3.2×10^{-3}	0.07×10^{-3}
P_3	0.025	1.1×10^{-3}	0.41×10^{-3}	0.04×10^{-3}	0.070	3.8×10^{-3}	2.7×10^{-3}	0.03×10^{-3}

Considerando el histograma de la Figura 4(b) como la mezcla a aproximar, los métodos son operados usando dos diferentes conjuntos de parámetros iniciales (mostrados en la Tabla 2). Al final del proceso de estimación, cada parámetro $\tilde{\theta}$ de la mezcla estimada es comparado con cada parámetro θ de la mezcla real (los cuales son mostrados en la Tabla 1). La diferencia entre ellos es modelada como el error de estimación E , el cual es expresado como $E = |\theta - \tilde{\theta}|$. La Tabla 2 resume los resultados obtenidos en el experimento, donde se muestra los valores promedio obtenidos durante 30 diferentes ejecuciones (el número de ejecuciones es realizada con el

objetivo de asegurar consistencia en los resultados). Del análisis de la Tabla 2 puede notarse como los algoritmos EM y LM presentan sensibilidad a la elección de sus valores iniciales. Tal hecho se debe a la incapacidad por parte de los algoritmos de evitar quedar atrapados en un mínimo local.

b) *Costo computacional*. Este índice de desempeño considera la medición del número de iteraciones, así como también el tiempo requerido por EM, LM y ABC para calcular los parámetros de la mezcla. Como criterio de paro se considero la iteración a partir de la cual ningún cambio en la mezcla fue detectado. En la comparación se usó el histograma de la Figura 4(b) como la mezcla a aproximar, considerando 3 clases. La Tabla 3 resume los resultados obtenidos, mostrándose los valores promedio obtenidos considerando 30 diferentes ejecuciones. Después de observar la Tabla 3, resulta evidente que el algoritmo EM utiliza mucho mayor numero de iteraciones para asegurar convergencia, mientras que el método LM es el que invierte mayor tiempo en encontrar la solución. Por otro lado ABC, resulta el enfoque con la mejor relación de desempeño.

Tabla 3: Comparación entre los algoritmos EM, LM y ABC, considerando el número de iteraciones y tiempo computacional.

Iteraciones	
Tiempo	4(b)
	1865
EM	2.81s
	975
LM	5.04s
	389
ABC	0.67s

4 Detección de círculos en imágenes digitales

La detección de círculos con ABC involucra una etapa de pre-procesamiento que requiere el marcado del contorno de los objetos presentes en la imagen. Para este propósito el algoritmo clásico Canny es usado para dicha tarea. Posteriormente, las posiciones de los píxeles borde encontrados son almacenadas dentro de un vector $P = \{p_1, p_2, \dots, p_{E_p}\}$, siendo E_p el número total de píxeles borde en la imagen.

4.1 Representación de cada individuo

Para poder construir cada círculo candidato C (fuente de alimento dentro del contexto de ABC), deben combinarse tres índices i_1, i_2 e i_3 que representan tres píxeles previamente almacenados en el vector P . Por lo tanto, cada fuente de alimento es codificada como $C = \{p_{i_1}, p_{i_2}, p_{i_3}\}$, el cual representa un círculo cuyos tres puntos caen sobre su propia circunferencia. Dicho círculo candidato es considerado como una solución potencial al problema de detección. Considerando la configuración de los puntos borde en la Figura 8, el centro del círculo (x_0, y_0) y el radio r de C puede ser calculados usando simples ecuaciones geométricas [31].

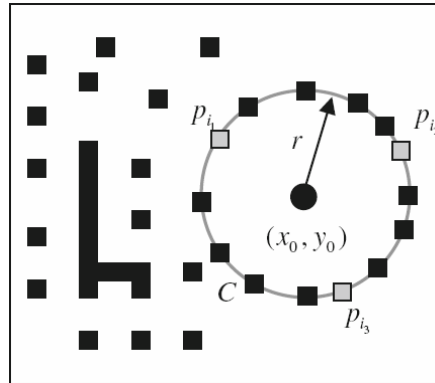


Figura 8. Círculos candidatos (representación individual) construidos a partir de la combinación de los puntos $P_{i_1}, P_{i_2}, P_{i_3}$.

4.2 Función objetivo

Para poder calcular el error en la coincidencia producida por el círculo candidato C , las coordenadas de su circunferencia son calculadas como una figura virtual que debe ser validada, *i.e.* confirmar si C existe en el mapa de bordes. Dichas coordenadas son agrupadas dentro del conjunto de prueba $S = \{s_1, s_2, \dots, s_{N_s}\}$ con N_s representando el número de puntos sobre los cuales la existencia de un punto borde correspondiente a C debe ser verificada.

En este enfoque el conjunto S es generado por el algoritmo de circunferencia de punto medio (MCA) [32]. MCA, es considerado el método más rápido proporcionando una precisión a nivel sub-píxel [32-34] para el cálculo digital de circunferencias. Considerando que $f_{\text{Circulo}}(x, y) = x^2 + y^2 - r^2$ define una primitiva circular, el MCA introduce un error e como una medida de la desviación de la posición del punto medio del píxel (distancia a nivel sub-píxel) caracterizado por $e = f_{\text{Circulo}}(x, y)$, siendo e cero para posiciones pertenecientes a la circunferencia, positivo para aquellos fuera de la misma y negativo para los que ocupan el interior. El error mínimo (*i.e.* el error mostrado por el píxel más cercano a la circunferencia ideal) es usado para decidir cual píxel debe ser colocado a continuación como la mejor aproximación para el borde de un círculo. Por otra parte, el tiempo computacional de MCA es reducido al considerar la simetría sobre los círculos. Para más detalles refiérase a [31].

La función objetivo $J(C)$ representa el error de coincidencia producido entre los píxeles S (calculados por MCA) del círculo candidato C y los píxeles que realmente existen en la imagen de bordes, dando como resultado:

$$J(C) = 1 - \frac{\sum_{v=1}^{N_s} E(s_v)}{N_s} \quad (14)$$

donde $E(s_v)$ es una función que verifica la existencia de un píxel en la posición s_v ($s_v = (x_v, y_v)$), con s_v perteneciendo a S y N_s siendo el número de píxeles pertenecientes al perímetro de C bajo prueba. Por lo tanto la función $E(s_v)$ queda definida como:

$$E(s_v) = \begin{cases} 1 & \text{si el píxel } (x_v, y_v) \text{ es punto borde} \\ 0 & \text{en otro caso} \end{cases} \quad (15)$$

Un valor de $J(C)$ cercano a cero significa una mejor coincidencia y representa una mejor cantidad de néctar dentro del marco del ABC. La Figura 9 muestra, mediante un ejemplo, el procedimiento para evaluar la función objetivo $J(C)$. Primero, tres puntos borde (como son mostrados en Figura 9a) codifican un círculo candidato C . Después,

usando el MCA una figura circular virtual es construida (véase Figura 9b). Dicha figura virtual, la cual es caracterizada por el vector S , agrupa un determinado número N_s de coordenadas de píxeles que son mostrados en la Figura 9b como un círculo formado por 56 casillas. Finalmente, la figura virtual es comparada con la imagen de bordes original punto a punto para poder encontrar coincidencias entre los píxeles borde y los virtuales. La Figura 9c muestra la operación de comparación entre el mapa de bordes original (Figura 9a) y la figura virtual (Figura 9b). Como resultado, sólo dieciocho píxeles borde están presentes en ambas imágenes, resultando:

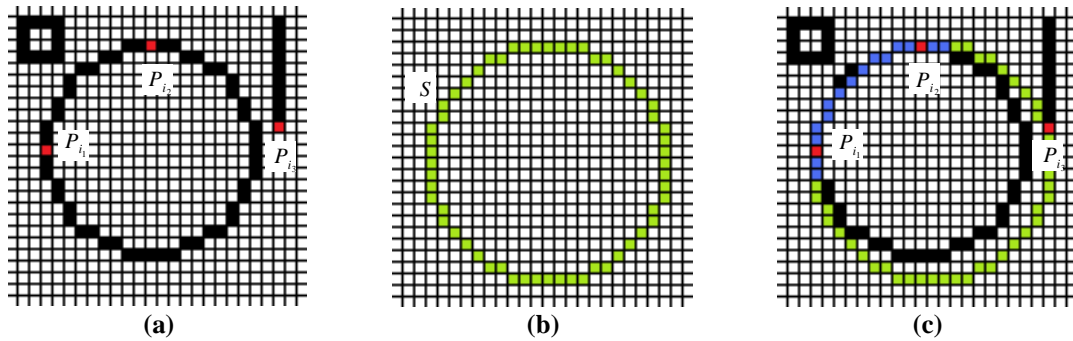
$$\sum_{v=1}^{N_s} E(x_v, y_v) = 18 \text{ y por lo tanto } J(C) \approx 0.67 .$$


Figure 9. Procedimiento para evaluar la función objetivo $J(C)$: (a) mapa de bordes original, (b) figura virtual generada por MCA considerando $C = \{p_{i_1}, p_{i_2}, p_{i_3}\}$. (c) Operación de comparación entre el mapa de bordes original mostrado en (a) y la figura virtual presentada en (b).

4.3 Resultados experimentales

Se realizaron diversas pruebas para evaluar el desempeño del detector de círculos propuesto. Tales experimentos refieren a las siguientes tareas:

1. Localización de círculos
2. Discriminación de formas
3. Aproximación circular: círculos ocluidos y detección de arcos

La Tabla 4 presenta los parámetros con los que se configuro el algoritmo ABC para todas las imágenes usadas en los experimentos.

Tabla 4: Parámetros del algoritmo ABC

N_p	L	NG
40	100	300

Localización de círculos

Imágenes sintéticas

Para el desarrollo experimental de esta prueba se usaron varias imágenes sintéticas de 320×240 píxeles, todas ellas contienen distintas cantidades de figuras circulares y algunas incluso han sido contaminadas agregando ruido a manera de incrementar la complejidad de la tarea de localización. El algoritmo fue ejecutado 100 veces para cada imagen de prueba, identificando exitosamente todos los círculos en la imagen. La detección probó ser robusta, requiriendo menos de un segundo en ser completada. La Figura 10 muestra el resultado después de aplicar el algoritmo sobre dos imágenes tomadas del conjunto experimental de imágenes.

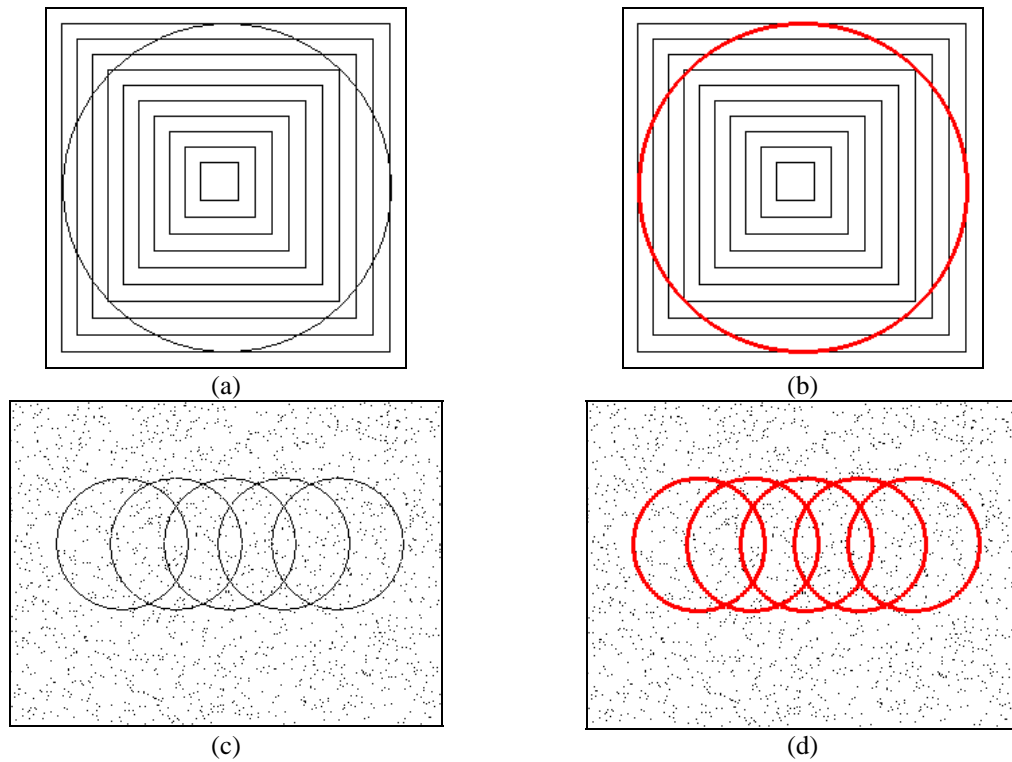


Figura 10. Localización de círculos sobre imágenes sintéticas. (a) Muestra la imagen original, mientras que (b) presenta el círculo detectado superpuesto. (c) Imagen contaminada con ruido sal y pimienta y (d) resultado superpuesto de manera distintiva.

Imágenes naturales

En este experimento se prueba la detección de círculos en imágenes reales. Para ello, se capturaron 25 imágenes de prueba usando una cámara digital. Las imágenes tienen una resolución de 640×480 píxeles y un formato de color de 8 bits. Cada escena incluye figuras circulares de complejidad variable. La Figura 11 muestra la detección de múltiples círculos en una imagen real del conjunto experimental.



Figura 11. Algoritmo de localización de círculos sobre imágenes reales. (a) Muestra la imagen original, mientras que (b) presenta los círculos detectados superpuestos.

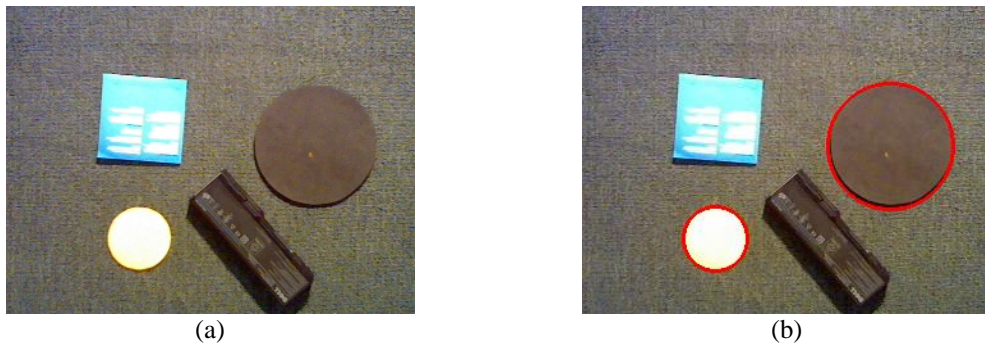


Figura 12. Discriminación de formas en imágenes reales. (a) Imagen original, (b) Círculos detectados superpuestos.



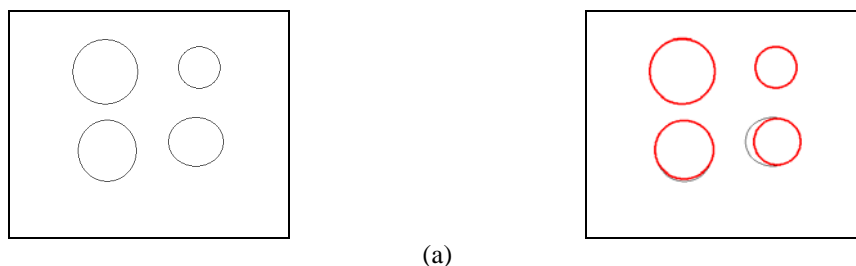
Figura 13. Discriminación de formas sobre imágenes sintéticas. (a) Imagen original contaminada con ruido sal y pimienta, (b) Círculos detectados superpuestos de forma distintiva.

Discriminación de formas

Esta sección discute la habilidad del detector para diferenciar patrones circulares, los cuales puedan estar presentes en la imagen, sobre cualquier otra figura. La figura 12 muestra cuatro diferentes figuras dentro de una imagen real de 320×240 píxeles. La figura 13 repite el experimento sobre una imagen sintética.

Aproximación circular: círculos ocluidos y detección de arcos

Una importante particularidad del detector basado en ABC, es que puede detectar círculos ocluidos o imperfectos, así mismo formas parcialmente definidas como segmentos de arco. La figura 14a muestra algunos ejemplos de aproximación circular. De la misma forma, el algoritmo propuesto es capaz de identificar los arcos o círculos ocluidos. Las figuras 14b y 14c muestran algunos ejemplos de esta funcionalidad. Un valor pequeño para $J(C)$, *i.e.* cercano a cero, refiere a un círculo mientras que un valor ligeramente mayor implica un arco o una figura circular ocluida.



(a)

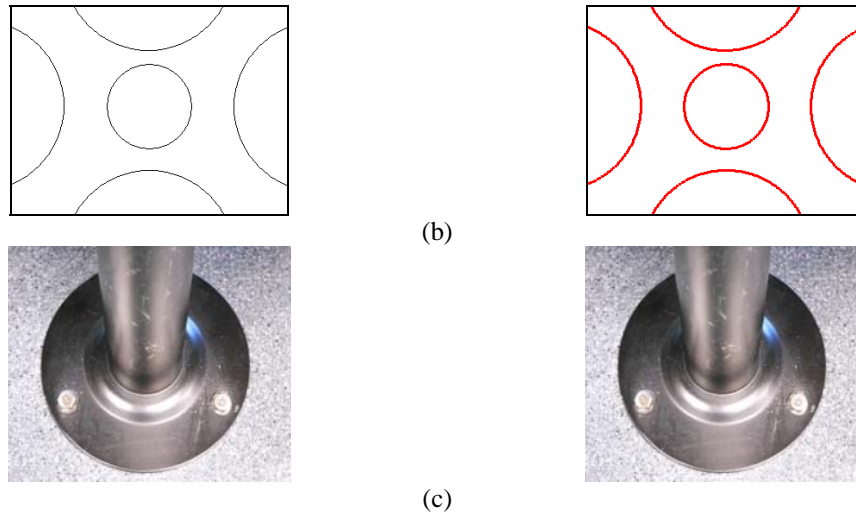


Figura 14. Detector basado en ABC aproximando figuras circulares y secciones de arco.

4.4 Comparación del rendimiento de detector ABC

Con el objetivo de analizar el desempeño del detector circular, el enfoque propuesto fue comparado con el detector basado en algoritmos genéticos (GA) y el algoritmo basado en la optimización de búsqueda de alimento de bacterias (BFOA). Mientras que el algoritmo ABC se configuro con los valores mostrados en la Tabla 3, los algoritmos GA y BFOA fueron implementados tal y como se sugiere en sus respectivas referencias [26,27].

Las imágenes difícilmente contienen círculos perfectos. Por consiguiente, para medir la precisión en la identificación, se coteja el resultado obtenido por la detección automática con el círculo “ideal”, determinado manualmente por una persona sobre la imagen de bordes. De esta manera si $(x_{ideal}, y_{ideal}, r_{ideal})$ representa los parámetros del círculo C_{ideal} determinado manualmente y (x_D, y_D, r_D) los del círculo C_D detectado de forma automática, el error de detección Es se define como:

$$Es = \eta \cdot (|x_{ideal} - x_D| + |y_{ideal} - y_D|) + \mu \cdot |r_{ideal} - r_D| \quad (16)$$

donde la diferencia del punto central $(|x_{true} - x_D| + |y_{true} - y_D|)$ mide el error de desplazamiento, mientras la desigualdad $(|r_{true} - r_D|)$ considera el error de tamaño. Los parámetros η y μ permiten ponderar de manera diferente la diferencia del punto central y de tamaño entre C_{ideal} y C_D en el error Es . En este trabajo, los parámetros η y μ fueron elegidos como 0.05 y 0.1 respectivamente. Tal configuración implica que la diferencia de tamaño tendrá un mayor peso en el error final Es , en comparación al desplazamiento. En el caso de que el valor de Es sea menor a 1, se considera que la detección tuvo éxito, de otra manera que fallo. Debe notarse que en el caso particular de $\eta = 0.05$ y $\mu = 0.1$, la condición de $Es < 1$ se produce, cuando la diferencia de radios no es mayor a 10 píxeles, mientras que el error de desplazamiento no excede los 20 píxeles. En general, la razón de éxito (RE), usada como índice de desempeño en este artículo, se define como el porcentaje de intentos en los que un algoritmo en particular es capaz de detectar ($Es < 1$) el círculo presente en la imagen.

La Figura 15 presenta el conjunto experimental de imágenes usadas en la comparación de los algoritmos. A diferencia de otras imágenes contenidas en bancos de prueba estándar, las imágenes de la Figura 15 fueron adquiridas o producidas especialmente para añadir complejidad en la detección circular, ya sea porque la imagen presenta un círculo difícilmente distinguible o porque se añadió ruido localmente en los bordes de la figura circular. Cabe mencionar que las imágenes de la Figura 15, también muestran el círculo detectado por el algoritmo ABC en cada uno de los casos.

En la comparación los tres métodos GA, BFAO y ABC fueron ejecutados considerando el conjunto experimental de las imágenes presentadas en la Figura 14. La Tabla 4 resume los resultados obtenidos en la comparación, considerando los índices de razón de éxito (RE), el promedio del error de detección (Es), los cuales fueron calculados considerando 35 diferentes ejecuciones sobre la misma imagen. Para facilitar la lectura de la Tabla 5, los mejores resultados fueron resaltados. Una inspección de la Tabla 5 revela que el método ABC es capaz de alcanzar los mejores índices de desempeño al presentar la más alta razón de éxito y el error de detección más pequeño.

Tabla 5: Resultados de los índices de desempeño, razón de éxito (RE) y error de detección (Es) de los algoritmos GA, BFOA y ABC, utilizando como conjunto experimental las imágenes mostradas en la Figura 15.

Imagen	Razón de éxito (RE) (%)			Promedio de $Es \pm$ desviación estándar		
	GA	BFOA	ABC	GA	BFOA	ABC
(a)	98	100	100	0.45±(0.022)	0.30±(0.033)	0.20±(0.021)
(b)	98	98	100	0.61±(0.022)	0.41±(0.034)	0.19±(0.035)
(c)	75	90	100	0.56±(0.029)	0.46±(0.051)	0.21±(0.012)
(d)	90	98	100	0.62±(0.021)	0.53±(0.018)	0.28±(0.018)
(e)	92	96	100	0.42±(0.019)	0.37±(0.011)	0.12±(0.039)
(f)	95	98	100	0.87±(0.056)	0.81±(0.021)	0.39±(0.027)

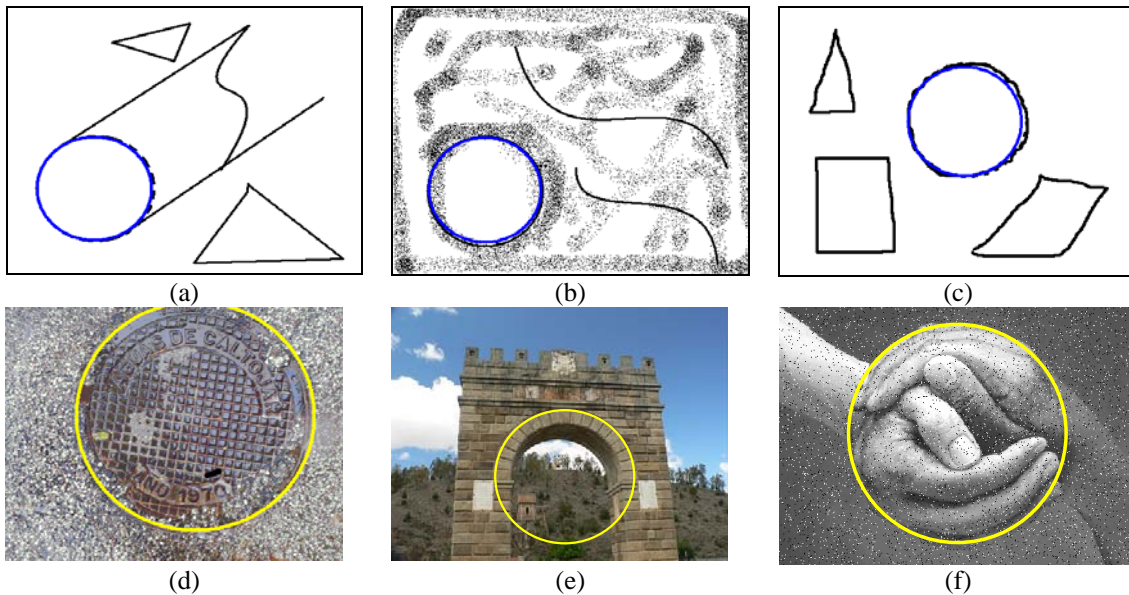


Figura 15. Conjunto experimental de imágenes usado en la comparación de los algoritmos GA, BFOA y ABC. Los círculos distintivos muestran los resultados de la detección obtenidos por ABC. (a)-(c) Imágenes sintéticas y (d)-(f) imágenes reales.

Para validar estadísticamente que el método ABC posee un mejor desempeño, se realizó un análisis no paramétrico conocido como la prueba de Wilcoxon [35] sobre el error de detección (E_s) producido por las pruebas 35 ejecuciones, que se reportaron en la Tabla 5. Dicho análisis realiza una prueba basada en la jerarquía, la cual coteja la diferencia en resultados entre dos métodos distintos. Wilcoxon considera como hipótesis nula que no existe diferencia entre los dos métodos bajo análisis, de lo contrario la hipótesis alternativa asume que si existe una diferencia. Los valores que arroja el análisis de Wilcoxon son los llamados valores- p , los cuales si son mas pequeños que 0.05 (que implica el nivel de certidumbre del 95%) ofrecen evidencia suficiente para indicar que la hipótesis nula es incorrecta, de tal manera que los dos métodos bajo prueba deben de ser considerados como diferentes. La Tabla 6 reporta los valores- p producidos por el análisis de Wilcoxon para las pruebas ABC vs. GA y ABC vs. BFOA.

Tabla 6. Valores- p producidos por el análisis de Wilcoxon en la comparación de los algoritmos.

Imagen	Valores- p	
	ABC vs. GA	ABC vs. BFOA
(a)	1.9456e-004	1.9234e-004
(b)	1.6341e-004	1.8892e-004
(c)	1.4562e-004	1.8648e-004
(d)	1.1261e-004	1.2165e-004
(e)	1.0179e-004	1.1930e-004
(f)	0.8922e-004	1.1683e-004

5 Conclusiones

En este artículo se propuso el uso del algoritmo de optimización “Artificial Bee Colony” (ABC) en el área de procesamiento digital de imágenes. Los problemas abordados fueron dos, segmentación automática y detección de círculos en imágenes digitales. Dichos problemas son considerados representativos dentro del análisis de imágenes y forman parte de la mayoría de los sistemas de visión. Bajo el enfoque propuesto, ambos problemas son abordados desde el punto de vista de la optimización.

En cuanto al problema de segmentación, el histograma de la imagen es aproximado por una mezcla de Gaussianas, cuyos parámetros que definen las clases de pixeles contenidos en la imagen, son determinados por el algoritmo ABC. Con el objetivo de mostrar las ventajas del método propuesto, se condujeron experimentos para comparar sus resultados con otros métodos populares para la estimación de mezclas Gaussianas, tales como el Expectación-Maximización [16] y el Levenberg-Marquadt [17]. Evidencia experimental mostro que el método ABC exhibe un mejor desempeño en términos de precisión y costo computacional, sin experimentar la típica sensibilidad a las condiciones iniciales.

Por otro lado, en la detección de círculos en imágenes se utiliza una combinación de tres puntos borde para la codificación de círculos candidatos. Utilizando las evaluaciones de una función objetivo (que determina si tales círculos están presentes en la imagen) el algoritmo ABC realiza una búsqueda eficiente hasta encontrar el círculo que mejor se aproxime a aquel contenido en la imagen. Se condujeron experimentos para comparar el enfoque propuesto con otros detectores basados en técnicas de computo evolutivo. Resultados experimentales validados por el análisis estadístico de Wilcoxon demostraron que el algoritmo ABC detecta círculos de una manera más robusta y precisa que sus contrapartes.

De la utilización del algoritmo ABC a los problemas planteados se puede concluir que los métodos tradicionales de procesamiento de imagen presentan diferentes dificultades, al momento de ser usados en imágenes que poseen ruido considerable y distorsiones. Bajo tales condiciones, el uso de ABC presenta un mejor rendimiento.

Referencias

- [1] Holland JH (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor..
- [2] Price K, Storn R, Lampinen A (2005) *Differential evolution a practical approach to global optimization*. In: Springer Natural Computing Series.
- [3] Szu, H, Chanyagorn P, Hwang W, Paulin M, Yamakawa T. Collective and distributive swarm intelligence: evolutionary biological survey. *International Congress Series*, 1269, 2004, 46-49.
- [4] Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *IEEE international conference on neural networks*, Piscataway, NJ, pp 1942–1948.
- [5] Dorigo M, Maniezzo V, Colomi (1991) A positive feedback as a search strategy. Technical report 91-016, Politecnico di Milano, Italy.
- [6] Liu Y, Passino K (2002) Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors. *J Optim Theory Appl* 115(3):603–628.
- [7] Karaboga D (2005) An idea based on honey bee swarm for numerical optimization, technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- [8] Guo R. and Pandit S.M., “Automatic threshold selection based on histogram modes and discriminant criterion,” *Mach. Vis. Appl.*, vol.10, pp.331–338, 1998.
- [9] Pal N.R. and Pal S.K., “A review on image segmentation techniques,” *Pattern Recognit.*, vol.26, pp.1277–1294, 1993.
- [10] Shaoo P.K., Soltani S., Wong A.K.C., and Chen Y.C., “Survey: A survey of thresholding techniques,” *Comput. Vis. Graph. Image Process.*, vol.41, pp.233–260, 1988.
- [11] Snyder W., Bilbro G., Logenthiran A., and Rajala S., “Optimal thresholding: A new approach,” *Pattern Recognit. Lett.*, vol.11, pp.803–810, 1990.
- [12] Chen S., Wang M., (2005) Seeking multi-thresholds directly from support vectors for image segmentation, *Neurocomputing*, 67(4), 335-344.
- [13] Chih-Chih L., (2006) A Novel Image Segmentation Approach Based on Particle Swarm Optimization, *IEICE Trans. Fundamentals*, 89(1), 324-327.
- [14] Cuevas E., Zaldivar D., Pérez-Cisneros M., Seeking multi-threshold for image segmentation with learning automata, *Machine Vision with Applications*, 22, (2011), 805-818.
- [15] González R.C. and Woods R.E., *Digital Image Processing*, Addison Wesley, Reading, MA, 1992.
- [16] Zhang, Z., Chen, C., Sun, J., Chan, L.: EM algorithms for Gaussian mixtures with split-and-merge operation, *Pattern Recognition*, 36, 1973-1983 (2003).
- [17] Park H, Amari S, Fukumizu K.: Adaptive natural gradient learning algorithms for various stochastic models. *Neural Networks*, 13,755–764 (2000).
- [18] Gupta, L., Sortrakul, T.: A Gaussian-Mixture-Based Image segmentation Algorithm, *Pattern Recognition*, 31(3), pp.315–325 (1998).
- [19] da Fontoura Costa L, Marcondes Cesar R Jr (2001) *Shape analysis and classification*. CRC Press, Boca Raton.
- [20] Muammar H, Nixon M (1989) Approaches to extending the Hough transform. In: *Proceedings of the international conference on acoustics, speech and signal processing ICASSP_89*, vol 3, pp 1556–1559.

- [21] Atherton TJ, Kerbyson DJ (1993) Using phase to represent radius in the coherent circle Hough transform. In: Proceedings of the IEE colloquium on the Hough transform. IEE, London.
- [22] Shaked D, Yaron O, Kiryati N (1996) Deriving stopping rules for the probabilistic Hough transform by sequential analysis. *Comput Vis Image Underst* 63:512–526.
- [23] Xu L, Oja E, Kultanen P (1990) A new curve detection method: randomized Hough transform (RHT). *Pattern Recogn Lett* 11(5):331–338.
- [24] Han JH, Koczy LT, Poston T (1993) Fuzzy Hough transform. In: Proceedings of the 2nd international conference on fuzzy systems, vol 2, pp 803–808. [2] Richard E. Korf. Iterative-deepening A*: an optimal admissible tree search. In *Proc. of the IX Int. Joint Conf. on Artificial Intelligence (IJCAI'85)*, pages 1034–1036, 1985.
- [25] Becker J, Grousson S, Coltuc D (2002) From Hough transforms to integral transforms. In: Proceedings of the international geoscience and remote sensing symposium 2002 IGARSS_02, vol 3, pp 1444–1446.
- [26] Ayala-Ramirez V, Garcia-Capulin CH, Perez-Garcia A, Sanchez-Yanez RE (2006) Circle detection on images using genetic algorithms. *Pattern Recogn Lett* 27:652–657.
- [27] Dasgupta S, Das S, Biswas A, Abraham A (2009) Automatic circle detection on digital images with an adaptive bacterial foraging algorithm. *Soft Comput.* doi:10.1007/s00500-009-0508-z.
- [28] Erik Cuevas, Noé Ortega-Sánchez, Daniel Zaldívar, Marco Pérez-Cisneros. Circle Detection by Harmony Search Optimization, *Journal of Intelligent & Robotic Systems* 66(3), (2012), 359-376.
- [29] Erik Cuevas, Daniel Zaldívar, Marco Pérez-Cisneros. Circle detection using discrete differential evolution optimization, *Pattern Analysis and Applications*, 14(1), (2011), 93-107.
- [30] Karaboga D., Akay B. A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation* 214 (2009) 108-132.
- [31] Van Aken JR (2005) Efficient ellipse-drawing algorithm. *IEEE Comput Graph Appl* 4(9):24–35.
- [32] Bresenham JE (1987) A linear algorithm for incremental digital display of circular arcs. *Commun ACM* 20:100–106.
- [33] Mairessea F, Sliwa T, Binczak S, Voisina Y (2007) Subpixel determination of imperfect circles characteristics. *Pattern Recogn* 41:250–271.
- [34] Bhowmick P, Bhattacharya BB (2008) Number-theoretic interpretation and construction of a digital circle. *Discrete Appl Math* 156:2381–2399.
- [35] Garcia S, Molina D, Lozano M, Herrera F (2008) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special session on real parameter optimization. *J Heurist.* doi:10.1007/s10732-008-9080-4.