Extracción de Contornos mediante Algoritmos Evolutivos

Román Katz y Claudio Delrieux

Departamento de Ingeniería Eléctrica y de Computadoras
Universidad Nacional del Sur - Av. Alem 1253, (8000) Bahía Blanca, ARGENTINA.
e-mail: claudio@acm.org

Resumen

La extracción de contornos es indispensable para una gran cantidad de tareas asociadas con el reconocimiento e identificación de patrones en imágenes digitales y la visión computacional. La mayoría de las técnicas de segmentación de contornos se basa en la detección de gradientes locales, por lo que con imágenes ruidosas estos métodos se vuelven inestables y poco confiables. Por lo tanto se requieren mecanismos globales que permitan sobreponerse adecuadamente a los mínimos locales generados por el ruido. En este trabajo proponemos el uso de algoritmos evolutivos como mecanismo heurístico para la extracción de contornos en imágenes con ruido. Los algoritmos evolutivos exploran el espacio combinatorio de posibles soluciones por medio de un proceso de selección de mejores soluciones (generadas por mutación *y* cruzamiento*) seguidas por la evaluación de la adecuación de las nuevas soluciones (*fitness*) y la* selección *de un nuevo conjunto de soluciones. Cada posible solución es un contorno, cuyo* fitness *es una medida de la diferencia de intensidades acumulada a lo largo del mismo. Este proceso se repite iterativamente a partir de una primera aproximación (la* población inicial*), ya sea un cierto número de generaciones o bien hasta alcanzar algún criterio conveniente de detención, por ejemplo encontrar un contorno cuyo fitness es adecuadamente bajo. La exploración uniforme del espacio de soluciones y el no estancamiento en mínimos locales (principalmente por efecto de la operación de* mutación*) inducen a una mejora gradual de los resultados con la evolución de las poblaciones.*

Palabras Clave: Extracción de Contornos, Reconocimiento de Patrones, Procesamiento de Imágenes, Algoritmos Evolutivos, Metaheurísticas.

1 Introducción

La extracción de contornos en imágenes digitales es una operación de gran interés en los procesos de segmentación e identificación de patrones, tanto para tareas de reconocimiento e interpretación, como también de clasificación de objetos [4]. El operador gradiente es una herramienta ampliamente utilizada a tal efecto, detectando las variaciones en los niveles de intensidad que pudieran corresponder a los contornos de interés. Aunque esta metodología brinda resultados aceptables para casos típicos, existe gran variedad de situaciones en las que se requiere un esfuerzo computacional adicional que permita ampliar su rango de aplicación. Tal es el caso de la extracción de contornos en imágenes ruidosas o con objetos con niveles de intensidad no uniforme. En estos casos se realiza típicamente una detección de bordes utilizando el operador gradiente, seguida por algún esquema de procesamiento global.

Una de las técnicas que actualmente se está investigando se denomina *contornos activos* [1, 6], y consiste en utilizar curvas inicializadas por el usuario, las cuales se mueven dentro de la imagen hasta encontrar el contorno buscado. Para ello se utilizan mecansimos diversos, como B-Splines, flujo del vector gradiente, etc. En general, los contornos activos poseen limitaciones respecto de las concavidades de las fronteras a segmentar [10].

Es este trabajo presentamos un sistema de extracción de contornos que combina el uso del operador gradiente en la implementación de un algoritmo evolutivo. El uso de algoritmos evolutivos [2] nos provee una herramienta de resolución capaz de encontrar soluciones próximas a la óptima a este problema de formulación matemática no trivial y de gran complejidad computacional. En la Sección 2 presentamos la problemática de extracción de contornos, con énfasis particular en imágenes con niveles de intensidad no uniformes y sometidas a ruido. En la Sección 3 se introduce los algoritmos evolutivos y su aplicación en la extracción de contornos. Finalmente en las Secciones 4 y 5 se presentan respectivamente ejemplos de aplicación con los resultados obtenidos y las conclusiones.

2 Extracción de Contornos

En general las técnicas utilizadas caracterizan a los contornos de un objeto mediante la detección de sus bordes en función de discontinuidades importantes entre los niveles de intensidad de pixels vecinos [4]. Es adecuada en este sentido la utilización del operador gradiente para encontrar los cambios bruscos de intensidad $I(x, y)$ de los objetos que conforman la imagen [3]. El mismo está descripto matemáticamente por la definición siguiente.

$$\nabla I(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix}. \tag{1}$$

En muchas ocasiones resulta conveniente discriminar el *gradiente escalar* $\nabla = \sqrt{G_x^2 + G_y^2}$ que abstrae la dirección del gradiente propiamente dicho. De ahora en adelante, cada mención al gradiente ∇ se refiere al gradiente escalar, salvo aclaración específica.

Los operadores G_x y G_y representan una implementación genérica de las derivadas direccionales en la imagen digital, pudiéndose implementar por medio de las "máscaras" de Roberts, Prewitt, o Sobel. En este trabajo adoptamos esta última posibilidad porque experimentalmente es la que brinda mejores resultados. Por lo tanto, si $I(x,y)$ denota la intensidad en el pixel (x,y), entonces la aplicación de los operadores mencionados en el pixel (x_i, y_j) es la siguiente.

$$\begin{aligned} G_x &= (I(x_{i-1}, y_{j-1}) + 2I(x_i, y_{j-1}) + I(x_{i+1}, y_{i-1})) - \\ &\quad (I(x_{i-1}, y_{j+1}) + 2I(x_i, y_{j+1}) + I(x_{i+1}, y_{i+1})) \end{aligned} \tag{2}$$

$$\begin{aligned} G_y &= (I(x_{i-1}, y_{j-1}) + 2I(x_{i-1}, y_j) + I(x_{i-1}, y_{j+1})) - \\ &\quad (I(x_{i+1}, y_{j-1}) + 2I(x_{i+1}, y_j) + I(x_{i+1}, y_{j+1})) \end{aligned}$$

Es fácil ver, entonces, que estos operadores se pueden encontrar por medio de la respectiva convolución de la imágen con las máscaras escalares

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ y } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

En la Fig. 1(b) se observa el gráfico de la magnitud del gradiente ∇ correspondiente a la imagen de la Fig. 1(a). En esta imagen el objeto principal posee un nivel de intensidad uniforme y se encuentra libre de ruido. Es por esto que el contorno del objeto se presenta altamente distinguible. Sin embargo, la presencia de ruido en las imágenes digitales es inevitable, tanto por los diferentes procesos intervinientes en su adquisición (por ejemplo el ruido de origen térmico en las cámaras digitales) como por los errores numéricos y de cuantización que ocurren en el almacenamiento y procesamiento digital.

Cuando las imágenes contienen ruido (como ocurre por ejemplo en la Fig. 1(c)) la magnitud ∇ continúa rescatando el contorno (Fig. 1(d)), aunque se percibe su notable degradación como consecuencia de la "amplificación" de las pequeñas perturbaciones locales debidas al ruido. Si además consideramos que los objetos que

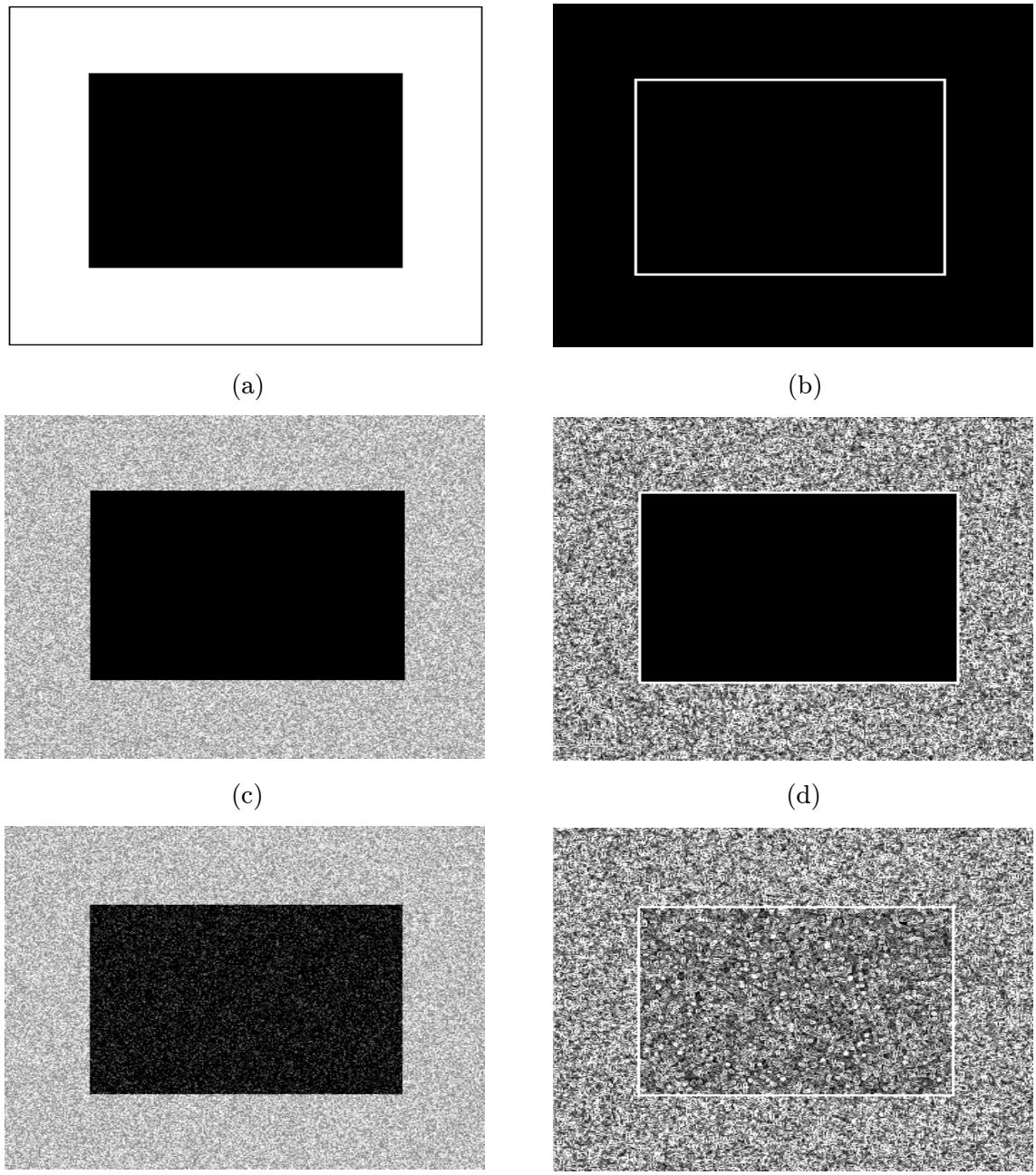


Figura 1: Imágenes de entrada en (a), (c) y (e) y sus correspondientes gráficos de ∇ en (b), (d) y (f) respectivamente.

componen la imagen pueden presentar niveles de intensidad variados (Fig. 1(e)), la percepción nítida del contorno mediante ∇ se dificulta notablemente (Fig. 1(f)). Es esta última una situación recurrente en la práctica, la intensidad de los objetos no tiene en general un nivel uniforme y los pixels pertenecientes al contorno de los objetos suelen estar afectados por discontinuidades espúrias producidas por el ruido.

La segmentación de contornos por medio de la utilización de técnicas de procesamiento local basados en la aplicación aislada del operador gradiente se ve notablemente restringida. Esto es así dado que este operador, al amplificar pequeñas diferencias locales en las imágenes con ruido, tiende a encontrar soluciones alejadas de la óptima. Si una medida de lo adecuado de un contorno es la diferencia acumulada entre las intensidades de sus pixels sucesivos, entonces este operador minimiza localmente esta diferencia, pero queda "estancado" en mínimos locales, mientras que el mínimo global sería el contorno que ajusta perfectamente el límite de los objetos que componen la imagen.

Luego las estrategias típicas empleadas bajo estas condiciones dificultosas suelen combinar detección de bordes y esquemas de procesamiento global tendientes a evitar los mínimos locales. Tal es el caso de los métodos compuestos de extracción de contornos mediante la detección de bordes junto con la transformada de Hough, la búsqueda en grafos [3] o la programación dinámica [4].

3 Algoritmos Evolutivos

3.1 Nociones Preliminares

Los algoritmos evolutivos [2] son métodos de búsqueda estocástica que permiten explorar un espacio de soluciones de gran tamaño en un tiempo aceptable, con el objeto de encontrar una solución que, sin ser necesariamente la óptima, está suficientemente próxima a ésta como para ser aceptable. Entre estos algoritmos se destacan la programación evolutiva (PE) [9] y la programación genética (PG) [5], las cuales son técnicas que surgen de imitar lo que conocemos de la evolución natural. La PE y la PG, entre otros contextos, son adecuadas para resolver problemas que con los métodos de Inteligencia Artificial de alto nivel tradicionales (búsqueda heurística, lógica, etc.) son o bien intratables o bien imprácticos, razón por la que en la actualidad son utilizados con éxito en diversidad de problemas de optimización combinatoria, scheduling, clasificación y identificación de sistemas y reconocimiento de patrones [7, 8].

Las ideas principales provienen de una metáfora de la evolución natural, en la cual existen individuos (fenotipos) que expresan una información genética (genotipo), y además están sujetos a la presión evolutiva del medio (*fitness*). Los individuos más exitosos (con mejor fitness) ven aumentadas sus posibilidades reproductivas, generando nuevos individuos con su mismo genotipo. Sin embargo, en el proceso reproductivo ocurre el fenómeno distintivo de la evolución natural: la aparición de mutaciones al azar. Este mismo esquema dinámico, con un sinnúmero de variaciones, ha sido desarrollado para diversas aplicaciones.

Un algoritmo evolutivo mantiene una *población* de soluciones posibles del problema y permite que las mismas progresen a través de la transformación de la *población* en las sucesivas generaciones. Operaciones de *mutación* y *cruzamiento* son las encargadas de producir la siguiente población a partir de la actual. El *cruzamiento* combina dos o más soluciones para generar una (o más) nuevas soluciones, mientras que la *mutación* obtiene una nueva solución perturbando una solución previa. En cada una de las generaciones se evalúa una función de fitness o costo para todas las soluciones, cuantificando la cercanía de los resultados al valor óptimo y efectuando una *selección* de las mejores para conformar la siguiente población.

Este proceso de mutación, cruzamiento, evaluación del fitness y selección se repite iterativamente a partir de una *población inicial*, ya sea un cierto número de generaciones o bien hasta alcanzar algún criterio conveniente de detención. La exploración uniforme del espacio de soluciones y el no estancamiento en mínimos locales (principalmente por efecto de la operación de mutación) inducen a una mejora gradual de los resultados con la progresiva evolución de las poblaciones. Finalmente la mejor solución de la población resultante, o la mejor solución de todo el proceso se elige como la solución encontrada.

3.2 Algoritmos Evolutivos en la Extracción de Contornos

Tal como se mostró en la Sección 2, la búsqueda de un contorno óptimo o cercano al óptimo a través del operador gradiente se ve radicalmente dificultada en presencia de ruido, debiéndose emplear técnicas de búsqueda global para evitar resultados inadecuados por convergencia en condiciones de mínimos locales. Resulta natural luego, conjugar la detección de bordes mediante el operador gradiente, con la aplicación de algoritmos evolutivos para alcanzar una solución del contorno próxima a la óptima global. Es en definitiva ésta la metodología utilizada en este trabajo: a partir de una población inicial de soluciones posibles del contorno, el sistema evoluciona mediante operaciones de mutación y cruzamiento, que inducen soluciones del contorno paulatinamente mejores. La selección se realiza evaluando el fitness de las soluciones mediante el operador ∇.

Supongamos la existencia de una población de C_N^K contornos, los cuales son los ancestros. Cada contorno $C_i (i \in N)$ posee un genotipo c_i codificado con números que representan los M puntos de la descripción poligonal del contorno. Entonces se crea un conjunto C_N^{K+1} de descendientes donde el genotipo c_i de cada contorno se obtiene a partir del genotipo anterior, aplicándole una mutación según la cual un solo punto del mismo es modificado aleatóriamente por medio de una perturbación con propiedades estadísticas determinadas. Considerando como ejemplo genérico la imagen de la Fig. 2(a), desarrollamos en las siguientes secciones los componentes fundamentales del algoritmo evolutivo implementado para la resolución de la problemática de detección de contornos.

3.2.1 Población Inicial

Sin dudas el contorno óptimo buscado (en nuestro caso de la Fig. 2(a)) estará ubicado en el límite externo del objeto de interés, como se puede observar en el gráfico del gradiente ∇ en la Fig. 2(b). Luego es posible generar una población inicial de soluciones adecuada para disparar nuestro algoritmo evolutivo que incluya este límite. En nuestro caso esto se efectúa posicionando manualmente dos círculos concéntricos (Fig. 2(c)), aunque sería posible determinar automáticamente estos radios en muchas situaciones.

Una vez determinados los radios mínimo y máximo, se generan aleatoriamente en la zona comprendida entre ambos una cantidad N de contornos conformada por igual cantidad M de puntos (Fig. 2(d)), espaciados en ángulos uniformes a lo largo del anillo y en posiciones aleatorias entre los radios mínimo y máximo. Es de destacar que es posible configurar tanto la cantidad de contornos a generar como la cantidad y el espaciamiento de los puntos que determinan cada uno de ellos.

3.2.2 Mutación y Cruzamiento

La operación de *mutación* implementada realiza, para cada uno de los contornos generados, la selección aleatoria de uno de sus puntos. Luego modifica su posición, con mayor incidencia sobre componente radial para una misma ubicación angular. De esa manera se obtiene un contorno modificado a partir del original. En las Fig. 3(a) y Fig. 3(b) se muestran respectivamente dos contornos y las modificaciones producidas por mutación para cada uno de ellos.

La operación de *cruzamiento* (*crossover*) se efectúa tomando los contornos de a pares. Luego se elige una posición aleatoria y se intercambian los puntos de ambas curvas desde dicho punto hasta el final del contorno, obteniendo dos nuevas curvas. Las Fig. 3(c) y Fig. 3(d) muestran respectivamente un par de contornos originales y los componentes permutados por efecto del *cruzamiento*.

3.2.3 Fitness y Selección

La evaluación del *fitness* de un contorno debe cuantificar cuánto aproxima el mismo al supuesto valor óptimo. Nuestro objetivo es detectar la poligonal de "costo mínimo", es decir, que minimice el costo local acumulado a lo largo de la misma. El costo local en cada uno de los puntos que componen el genotipo de cada contorno disminuye cuando se encuentra en una zona de la imagen donde localmente el gradiente es alto. Por dicha razón, utilizamos la siguiente función de costo local $k(x, y)$ en un pixel (x, y).

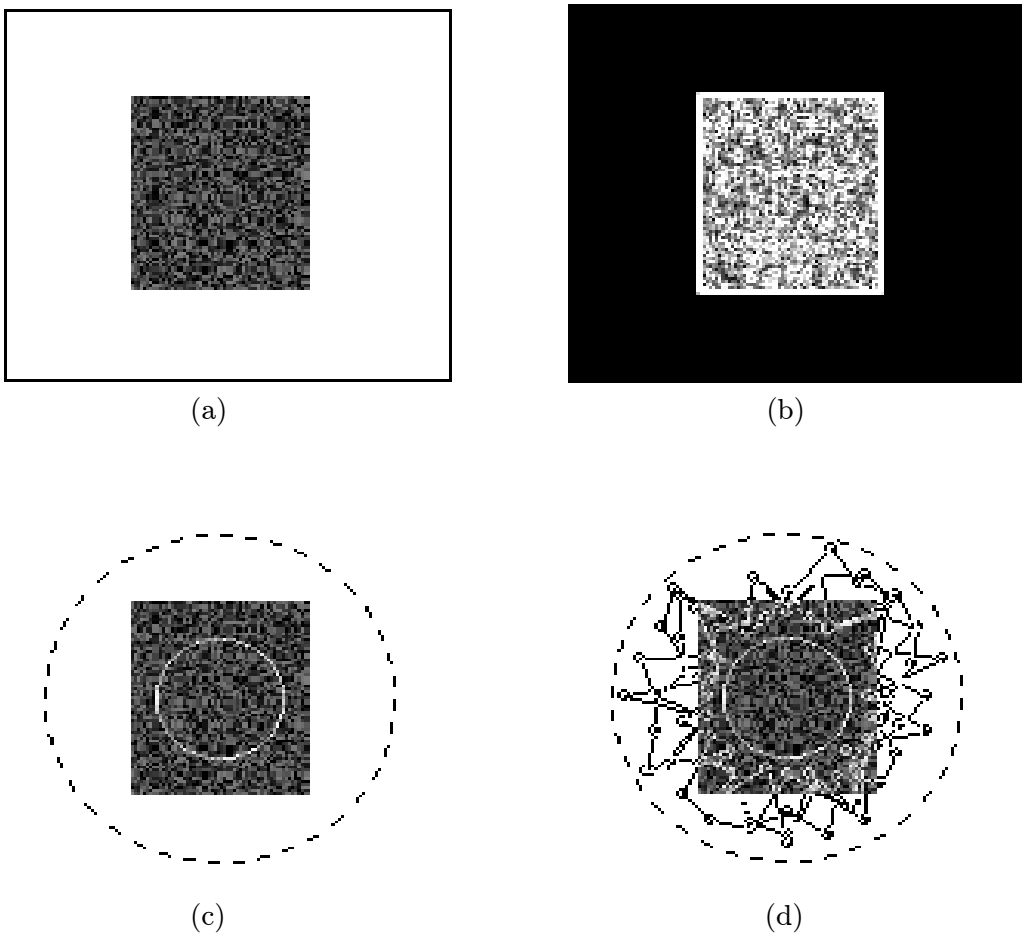


Figura 2: Ejemplo genérico de detección: imagen de entrada en (a), gráficos de ∇ en (b), círculos para generar la población inicial de contornos en (c) y contornos generados en (d).

$$k(x,y) = 9 - \sum_{i=-1}^{1} \sum_{j=-1}^{1} \nabla(x+1, y+j)$$

De esa manera, el costo local en un punto c_i^j del genotipo de un contorno C_i puede expresarse como $k(c_i^j)$.

El uso de máscaras de mayor tamaño implicarían un horizonte de mayor precisión, pero también un costo computacional mayor. También pueden utilizarse máscaras con funciones ponderadoras, que privilegien ciertas direcciones a costa de otras, etc. Es importante destacar que este cómputo se realiza solamente con los puntos que conforman el genotipo de un posible contorno en la población, y no con todos los pixels del mismo.

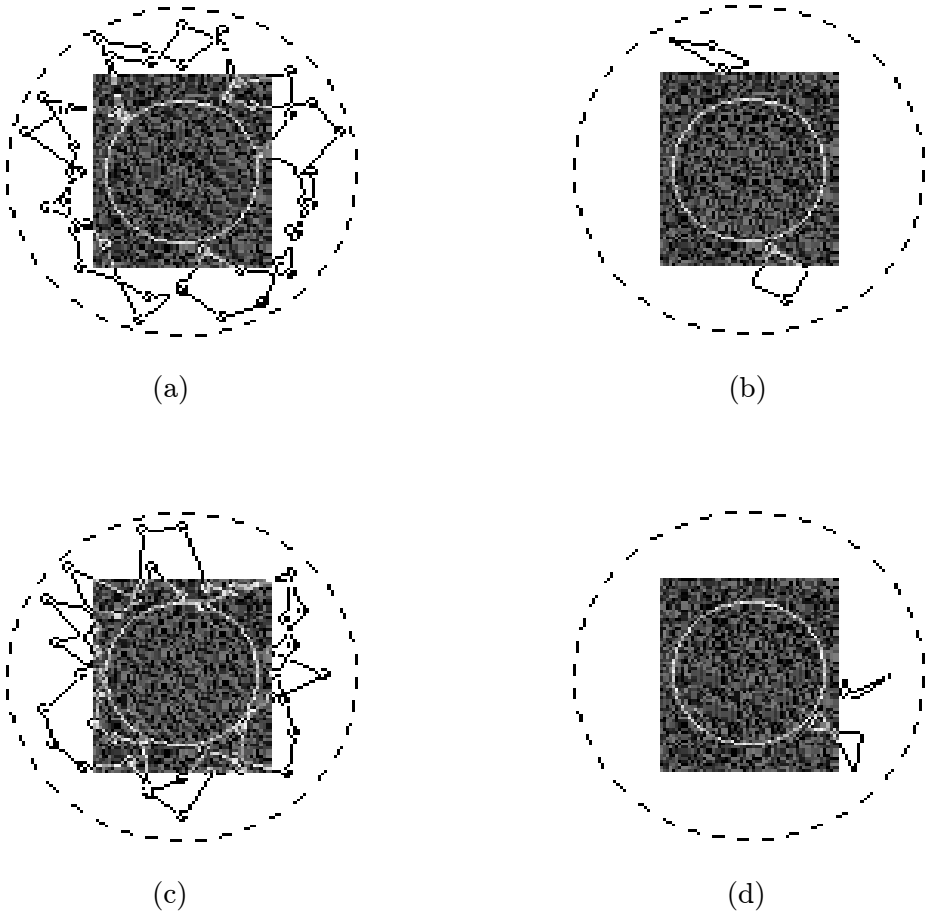


Figura 3: Mutación en (b) de los contornos mostrados en (a) y cruzamiento en (d) de los contornos mostrados en (c).

Para evaluar el fitness $f(C_i)$ del contorno i-ésimo simplemente es necesario acumular el costo local de cada uno de los M puntos que conforman su genotipo.

$$f(C_i) = \sum_{j=1}^{M} k(c_i^j)$$

En este trabajo, utilizamos como estrategia de selección la eliminación de los contornos generados cuyo costo superaba un cierto umbral porcentual en el rango entre el mínimo y el máximo de cada generación. Los rangos empleados exitosamente fueron elevados, manteniendo un nivel de supervivencia alto que nos permitió conservar una población prácticamente constante, al tiempo que nos posibilitó descartar aquellas soluciones radicalmente alejadas de la óptima.

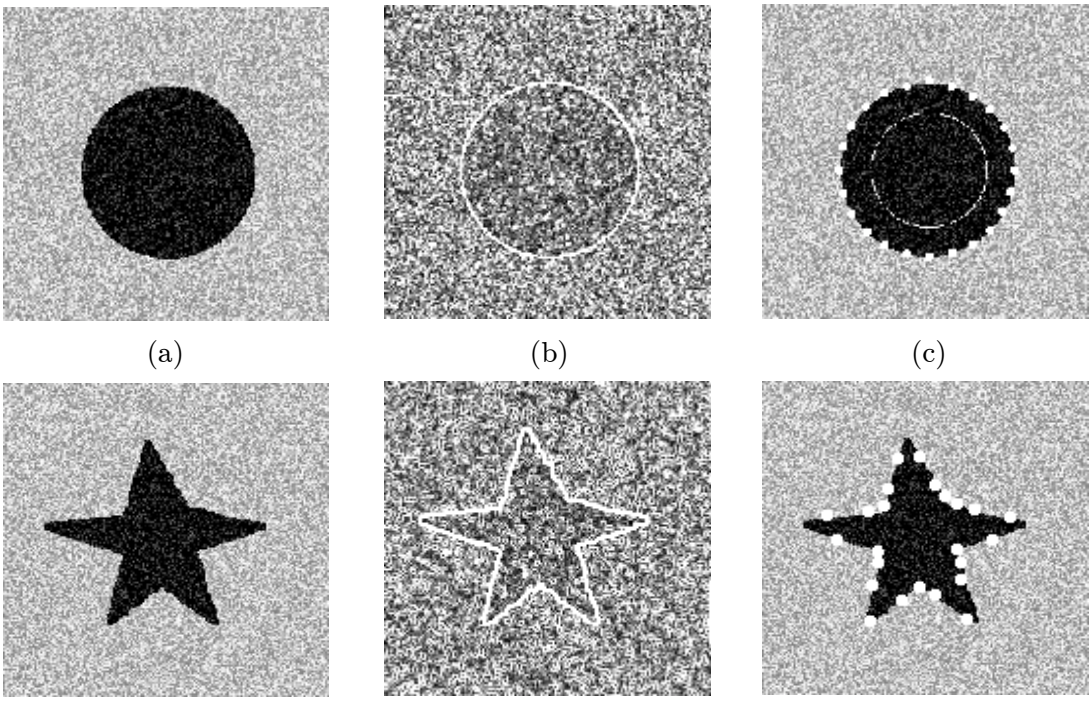


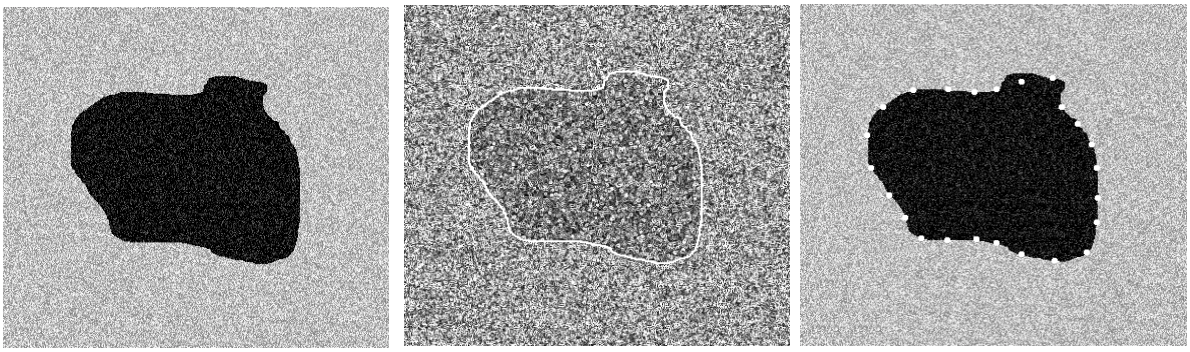
Figura 4: Imágenes de evaluación de entrada en (a) y (d), gráficos correspondientes de ∇ en (b) y (e) y del contorno detectado en (c) y (f).

4 Resultados Experimentales

Para evaluar el desempeño del sistema de extracción de contornos mediante algoritmos evolutivos se generaron imágenes ruidosas sumando ruido gausiano de media 0 y distintos desvíos estándar sobre diferentes imágenes de prueba.

La Fig. 4 muestra imágenes de entrada con ruido gausiano de desvío estándar σ=50, junto con el gráfico de su ∇ y el correspondiente contorno final encontrado. En el caso del objeto de la Fig. 4(a), el número de contornos utilizados fue de 3000 y el número de iteraciones de 15000, mientras que para el objeto de Fig. 4(d) la cantidad de iteraciones ascendió a 100000, destacándose que el tamaño de ambos objetos es comparable y que la diferencia en la cantidad de iteraciones requerida se debe a la diferencia en complejidad del objeto. Para alcanzar una detección de contorno correcta en objetos de mayor tamaño Fig. 5 se requirió un número de contornos más elevado (4000) como también una mayor cantidad de iteraciones (140000).

Los resultados alcanzados fueron en todos los casos altamente satisfactorios, detectando contornos muy aproximados a los óptimos, en condiciones diversas de ruido, tamaño y complejidad en los objetos. Para la detección en objetos de mayor



(a) (b) (c)

Figura 5: Evaluación en objetos de mayor tamaño: (a) imagen de entrada, (b) gráfico de ∇ y (c) contorno detectado.

tamaño o complejidad, el número de contornos debió elevarse, como también la cantidad de iteraciones requeridas para alcanzar buenos resultados.

Es de destacar que los mejores resultados en todas las experiencias realizadas se obtuvieron con probabilidades de mutación baja (menores a 10%), y que las soluciones finales obtenidas y mostradas corresponden, en todos los casos, a la mejor de todas las soluciones encontradas para cada una de las generaciones.

5 Conclusiones

Es este trabajo presentamos una metodología de extracción de contornos de objetos en imágenes digitales que combina el uso del operador gradiente en la implementación de un algoritmo evolutivo. El esquema desarrollado conjuga la simplicidad conceptual de la estrategia de detección provista por el operador gradiente y un comportamiento altamente robusto producto del mecanismo evolutivo. Esta robustez se manifiesta frente a distintas posibilidades en las imágenes de entrada, tanto en la forma y tamaño de los objetos de interés como también en condiciones de detección difíciles producidas por niveles de intensidad no uniformes y/o perturbaciones de ruido. El sistema respondió adecuadamente a diversas imágenes de evaluación utilizadas, encontrando contornos correctos, muy próximos al óptimo y en tiempos razonables.

La propuesta facilita la posible utilización y evaluación de otras alternativas en los operadores de *mutación* y *cruzamiento*, como también la posibilidad de adaptar diferentes esquemas para efectuar el computo del *fitness* (otros operadores frecuenciales, transformadas, etc.), manteniendo sin cambios el resto del algoritmo. La naturaleza "adaptativa" del sistema evolutivo permite, además, la potencial extensión

del sistema ampliando su campo de aplicación para soportar condiciones dinámicas de las imágenes de entrada.

Referencias

[1] A. Blake and M. Isard. *Active contours: the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion.* Springer-Verlag, 1998.

[2] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, New York, 1989.

[3] Rafael González and Richard Woods. *Digital Image Processing.* Addison-Wesley, Wilmington, USA, 1996.

[4] Anil Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, Cambridge, 1996.

[5] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, second edition, 1994.

[6] J. Park and J. M. Keller. Snakes on the Watershed. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1201–1205, 2001.

[7] D. Pham and D. Karaboga. *Intelligent Optimization Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing, and Neural Networks.* Springer-Verlag, New York, 1998.

[8] V. J. Rayward-Smith, I. H. Osman, and C. R. Reeves. *Modern Heuristic Search Methods.* Wiley, New York, 1996.

[9] I. Rechenberg. *Evolution strategie: Optimierung technischer systeme nach prinzipien der biologischen evolution.* Frommann-Holzboog, 1973.

[10] C. Xu and J. L. Prince. Snakes, Shapes, and Gradient Vector Flow,. *IEEE Transactions on Image Processing*, 28(3):359–369, 1998.