

Estrategias Evolutivas para la Detección de Contornos en Imágenes Digitales

Román Katz y Claudio Delrieux

*Universidad Nacional del Sur, Alem 1253, (8000) Bahía Blanca, ARGENTINA.
Voice: (54)(291)4595101 ext. 3381 — Fax: (54)(291)4595154 — e-mail: claudio@acm.org*

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 los cuales 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* [5, 8], 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 mecanismos diversos, como B-Splines [1], flujo del vector gradiente, etc. En general, los contornos activos poseen limitaciones respecto de las concavidades de las fronteras a segmentar [9].

En este trabajo presentamos un resumen del sistema y de los resultados obtenidos en la detección de bordes utilizando el uso combinado del operador gradiente con estrategias evolutivas [6, 7]. La utilización 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.

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:

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

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.

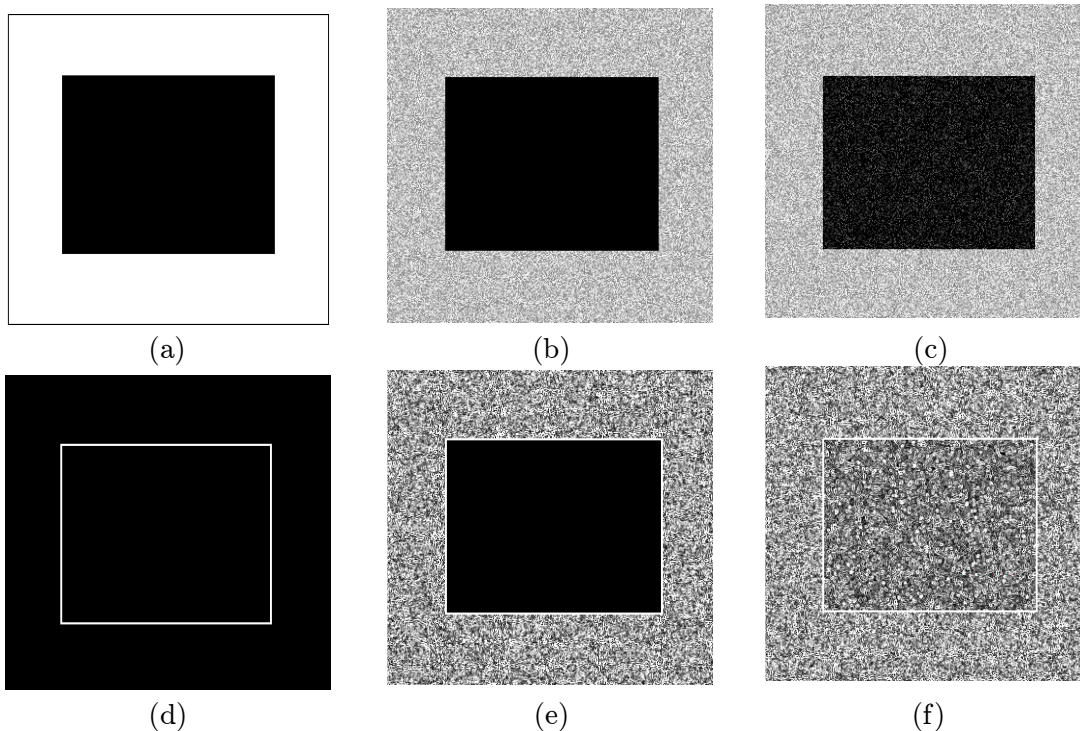


Figura 1: Imágenes de entrada en (a), (b) y (c) y sus correspondientes gráficos de ∇ en (d), (e) y (f) respectivamente.

En este trabajo adoptamos esta última posibilidad porque experimentalmente es la que brinda mejores resultados.

En la Fig. 1(d) se muestra 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 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(b)) la magnitud ∇ continúa rescatando el contorno (Fig. 1(e)), 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 componen la imagen pueden presentar niveles de intensidad variados (Fig. 1(c)), la percepción nítida del contorno mediante ∇ se dificulta notablemente (Fig. 1(f)), siendo esta una situación recurrente en la práctica.

La segmentación de contornos por medio de técnicas de procesamiento local del operador gradiente se ve notablemente restringida, dado que al amplificar pequeñas diferencias locales en las imágenes con ruido, se 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

Un algoritmo evolutivo [2] 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 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 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.

3.1 Algoritmos Evolutivos en la Extracción de Contornos

Luego de lo mostrado en la Sección 2 resulta natural 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 ésta en definitiva la metodología utilizada: 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 ∇ .

Población Inicial. Sin dudas el contorno óptimo buscado estará ubicado en el límite externo del objeto de interés. 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, aunque sería posible determinar automáticamente estos radios en muchas situaciones. Posteriormente se generan aleatoriamente en la zona comprendida entre ambos radios una cantidad N de contornos conformada por igual cantidad M de puntos, espaciados en ángulos uniformes a lo largo del anillo y en posiciones aleatorias de forma radial. Es posible en este punto configurar tanto la cantidad de contornos a generar como la cantidad y el espaciamiento de los puntos que determinan cada uno de ellos.

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. 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.

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 disminuye cuando se encuentra en una zona de la imagen donde localmente el gradiente es alto. Por dicha razón, utilizamos la función de costo local $k(x, y) = 9 - \sum_{i=-1}^1 \sum_{j=-1}^1 \nabla(x + 1, y + j)$ en un pixel (x, y) .

De esa manera, el costo local en un punto c_i^j del 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. 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 lo conforman $f(C_i) = \sum_{j=1}^M k(c_i^j)$.

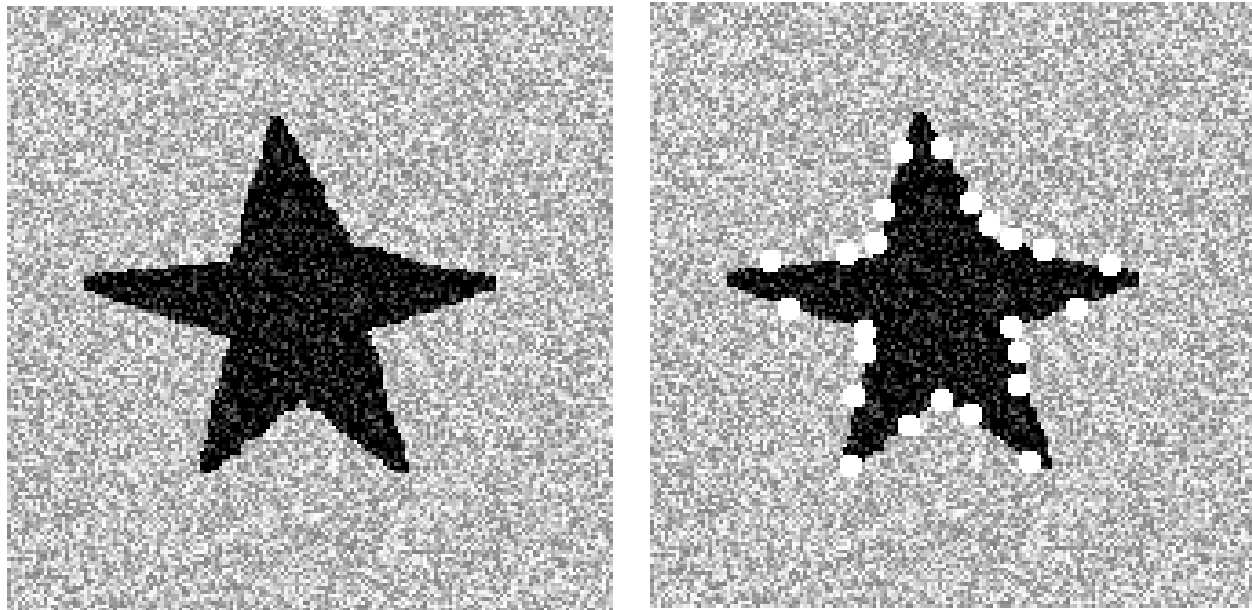


Figura 2: Imagen de evaluación con ruido Gaussiano y el contorno detectado.

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 permitió descartar aquellas soluciones radicalmente alejadas de la óptima.

4 Resultados Experimentales y Conclusiones

Para evaluar el desempeño del sistema de extracción de contornos mediante algoritmos evolutivos se generaron imágenes ruidosas sumando ruido gaussiano de media 0 y distintos desvíos estándar sobre diferentes imágenes de prueba. La Fig. 2 muestra una de entrada con ruido gaussiano de desvío estándar $\sigma=50$, y el correspondiente contorno final encontrado. La población de contornos utilizada en este ejemplo fue de 3000 y el número de iteraciones de 100000, siendo la convergencia razonablemente rápida.

En la Fig 3 se muestra un ejemplo de mayor dificultad, dado que se trata de un recorte de una imagen SAR, en la cual el ruido presente no es aditivo [1]. En esta imagen se eligió una de las zonas para inicializar la población inicial de 3500 contornos, los cuales luego de 500000 iteraciones alcanzaron la solución final mostrada. Para alcanzar una detección de contorno correcta en objetos de mayor tamaño se requirió en general un número de contornos más elevado como también una mayor cantidad de iteraciones.

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 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.

La estrategia 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, transforma-

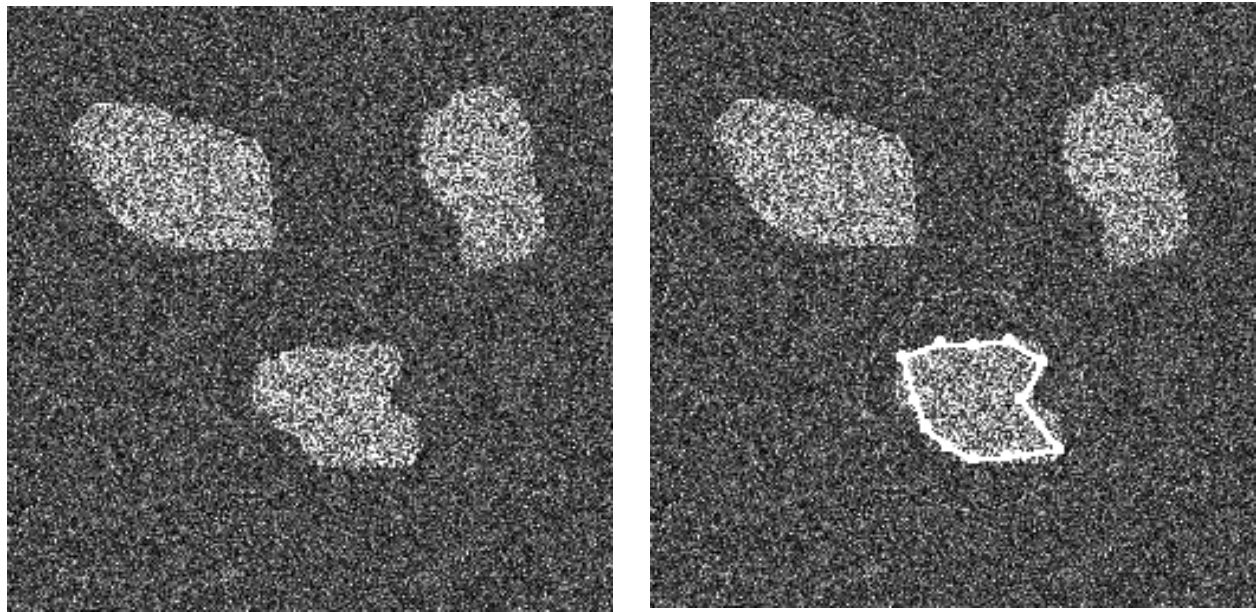


Figura 3: Imagen de evaluación con ruido no aditivo y el contorno detectado (imagen cortesía de J. Gambini).

das, 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] Juliana Gambini, Julio Jacobo, Marta Mejail, y Alejandro Frery. SAR image segmentation using B-Spline deformable contours. En *Congreso Argentino de Ciencias de la Computación*, págs. 503–510, Buenos Aires, 2002. VIII CACiC.
- [2] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York, 1989.
- [3] Rafael González y Richard Woods. *Digital Image Processing*. Addison-Wesley, Wilmington, USA, 1996.
- [4] Anil Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Cambridge, 1996.
- [5] M. Kass, A. Witkin, y D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1:321–331, 1987.
- [6] R. Katz y C. Delrieux. Extracción de Contornos Mediante Algoritmos Evolutivos. En *Congreso Argentino de Ciencias de la Computación*, , págs. 588–599. CACIC 2002, 2002.
- [7] R. Katz y C. Delrieux. Boundary Extraction Through Gradient-Based Evolutionary Algorithm. *Journal of Computer Science and Technology*, 3:7–12, 2003.
- [8] J. Park y J. M. Keller. Snakes on the Watershed. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1201–1205, 2001.
- [9] C. Xu and J. L. Prince. Snakes, Shapes, and Gradient Vector Flow,. *IEEE Transactions on Image Processing*, 28(3):359–369, 1998.