

Algoritmos para Generar Esqueletos K-Conectados Basados en Criterios de Conectividad

J. L. Díaz, C. Yáñez, G. Guzmán
Laboratorio de Procesamiento Digital de Imágenes
Centro de Investigación en Computación - IPN
07738 México, Distrito Federal

Tel. (52) 57 29 60 00 Ext. 56577 jdiaz@cic.ipn.mx, cyanez@cic.ipn.mx, jguzmanl@cic.ipn.mx

RESUMEN

En este artículo se presentan nuevos algoritmos de adelgazamiento que permiten generar esqueletos k -conectados de una imagen digital binaria, para conectividades de $k = 4, 6$ y 8 . Las ventajas que presentan dichos algoritmos a los desarrollados actualmente, es que no están orientados a caracteres escritos o algún tipo de objeto en particular, con lo cual, se pueden adaptar fácilmente a una aplicación cualquiera y generar buenos resultados, así como la flexibilidad de usar diferentes tipos de conectividad en el esqueleto resultante, ya que algunos algoritmos recientes trabajan con conectividad de tipo 8 y en algunos casos de tipo 4 . Adicionalmente, no se ven afectados seriamente por la presencia de ruido estructural en el contorno de los objetos a procesar.

Palabras clave:

Adelgazamiento, esqueletizado, máscaras.

I. INTRODUCCIÓN.

La esqueletización es una técnica importante usada en muchas áreas dentro del procesamiento digital de imágenes. Su propósito es reducir un objeto R de una imagen para generar un objeto S (llamado comúnmente esqueleto) cuyo grosor es de 1 píxel únicamente. Es bien sabido que las características topológicas generales basadas en conectividad no son suficientes para describir un objeto dado, debido a que el esqueleto de un objeto conectado simple puede ser un solo punto, como fue planteado por Rosenfeld [1], lo cual no es una representación útil en muchas aplicaciones. En consecuencia, es recomendable tomar en cuenta propiedades geométricas de los objetos, por ejemplo la condición de punto final de Pavlidis [2] que ayuda a preservar la "forma geométrica" del objeto en el esqueleto final.

Actualmente existen dos tipos de algoritmos para obtener el esqueleto de una imagen: algoritmos basados en píxeles y los no basados en píxeles [3]. En un método basado en píxeles, cada píxel de fondo participa en el cómputo del esqueleto

resultante. Dentro de las técnicas que hacen uso de este método se encuentran los algoritmos de adelgazamiento [4, 5] y las transformadas de distancia [4, 6]. La mayoría de los métodos más eficientes están basados en técnicas de adelgazamiento. En un método basado en una transformada de distancia, cada punto del esqueleto es un píxel de objeto con una distancia local máxima al fondo de la imagen. Por último, los métodos no basados en píxeles tratan de determinar analíticamente puntos simétricos del objeto, mediante algún criterio específico.

El principio básico de los métodos de adelgazamiento es eliminar repetidamente aquellos puntos del contorno de un objeto de tal manera que la eliminación de dicho punto no afecte la conectividad del objeto y respete la condición de punto final local, tal como se hace en el algoritmo clásico de Pavlidis [2]. El decidir cuando un píxel altera o no la conectividad global del objeto al ser eliminado, se puede determinar haciendo uso de criterios de conectividad locales, considerando la vecindad de dicho punto. Dichos criterios son formulados usualmente en términos de geometría digital [7], o por medio de máscaras [8].

En general, el adelgazamiento no permite la reconstrucción del objeto original, y aunque esta técnica preserva la conectividad, existen dos problemas, que son los focos de estudio actualmente: reducir el tiempo de ejecución de los algoritmos y la forma resultante del objeto. El primer punto no es tan crítico, considerando los avances notorios que se han logrado en las velocidades de los dispositivos usados en una computadora o se puede hacer uso de algoritmos paralelos como el desarrollado por Zhang y Suen [9]. Para el segundo punto, un problema que se puede tener con la forma del esqueleto es la generación de ramas parásitas que no están presentes en el objeto original. En este punto se encuentran los trabajos publicados por Sossa [10], Lü y Wang [11], Abdulla, Saleh y Morad [12] entre otros.

propósito del presente trabajo es presentar nuevos algoritmos de esqueletizado, que son básicamente de adelgazamiento secuencial. Se apoyan en el uso de máscaras que permiten determinar si un pixel debe o no ser borrado respetando la condición de conectividad y que no sea un pixel final. Estos algoritmos requieren de poco tiempo de procesamiento ya que reducen el número de operaciones aritméticas porque requieren en su mayor parte, de operaciones lógicas.

El documento está organizado de la siguiente manera: en la sección 2 se citan algunos conceptos importantes y se formula la estrategia general de los algoritmos de adelgazamiento propuestos. En la sección 3 se deducen las máscaras que se emplean en cada uno de los algoritmos según el tipo de conectividad a emplear. En la sección 4, se presentan las pruebas del sistema y finalmente en la sección 5 se citan las conclusiones del presente trabajo.

II. PRELIMINARES Y ESTRATEGIA GENERAL

Primeramente, citaremos algunos conceptos importantes de la geometría digital. Sean $p = (x, y)$, $q = (u, v)$ puntos de Z^2 . Las siguientes métricas en Z^2 son ampliamente usadas:

$$\begin{aligned} d_4 &= |x-u| + |y-v|, \\ d_8 &= \max(|x-u|, |y-v|), \\ d_{6R} &= \max(|x-u|, |y-v|, |x-u+y-v|), \\ d_{6L} &= \max(|x-u|, |y-v|, |x-u-y+v|). \end{aligned} \quad (1)$$

Cada una de estas métricas $d_k, k \in \{4, 6L, 6R, 8\}$, permiten llegar a una relación simétrica no reflexiva N_k en Z^2 definida por $(p, q) \in N_k \Leftrightarrow d_k(p, q) = 1$. La estructura (Z^2, N_k) se llama usualmente el grafo k -vecindado y los puntos p, q se dicen que son k -vecinos si $(p, q) \in N_k$, y adicionalmente $N_k(p)$ denota el conjunto de todos los k -vecinos de p . El concepto de conectividad de la teoría de grafos indirectos se aplica en el grafo (Z^2, N_k) . Luego, un subconjunto $R \subseteq Z^2$ está k -conectado si para $p, q \in R$ arbitrarios, existe una secuencia de puntos de $R, p = a_1, a_2, \dots, a_n = q$, donde a_i y a_{i+1} son k -vecinos para $i = 1, \dots, n-1$.

Ahora, sea R un subconjunto k -conectado de Z^2 . Un punto p de R se llama punto final de R si tiene únicamente un k -vecino en R . Se debe notar que el

único elemento de un conjunto de punto simple R es un punto final; y en el caso de que R tenga al menos dos puntos, entonces $p \in R$ es un punto final de R si p tiene exactamente un k -vecino en R . La idea del adelgazamiento es remover iterativamente puntos de R preservando no solamente la conectividad, sino también la condición de puntos finales en el conjunto actual de puntos en cada iteración. Es claro que los puntos de contorno de R son candidatos para ser eliminados sin alterar la conectividad. Recordando, p es un punto de contorno de R si p tiene un k' -vecino q con la condición de que $q \notin R$, donde $k' = 8$ para $k = 4$, $k' = 4$ para $k = 8$ y $k' = k$ para $k \in \{6L, 6R\}$. Debido a la definición de contorno de R se sigue conservando que el contorno es k -conectado.

La estrategia general de los algoritmos de adelgazamiento es la siguiente:

1. $R_0 = R; i = 1; R_i := R$.
2. Determinar el contorno C de R_i .
3. Para cada $p \in C$: Si p satisface la k -condición " p no es un punto final de R_i y $R_i \setminus \{p\}$ está k -conectado", entonces $R_i = R_i \setminus \{p\}$. En cualquier caso se debe repetir el paso 3 con el resto de puntos que integran el contorno.
4. Si $R_i = R_{i-1}$, entonces se detiene el algoritmo; R_{i-1} es el k -esqueleto de R . En caso contrario, incrementar i en 1, hacer $R_i = R_{i-1}$ e ir al paso 2.

III. MÁSCARAS DE ADELGAZAMIENTO.

Ahora se deducirán las máscaras para el conjunto T_4 . Supongamos que R debe ser un subconjunto 4-conectado de (Z^2, N_4) . Las plantillas que se están buscando deben cubrir todas las configuraciones locales posibles, donde la conectividad de tipo debe preservarse así como también los puntos finales. Inicialmente, necesitamos al menos plantillas las cuales se muestran en la Figura 1.

El símbolo X usado a lo largo del documento, indica que se trata de una condición de no importa, por cual, resulta indistinto si la intensidad de dicho pixel es de fondo o de imagen.

$$P_1 = \begin{bmatrix} X & 0 & X \\ 1 & p & 1 \\ X & 0 & X \end{bmatrix}, P_2 = \begin{bmatrix} X & 1 & X \\ 0 & p & 0 \\ X & 1 & X \end{bmatrix}, P_3 = \begin{bmatrix} X & X & X \\ X & p & X \\ 0 & 1 & 0 \end{bmatrix}$$

$$P_4 = \begin{bmatrix} X & X & 0 \\ X & p & 1 \\ X & X & 0 \end{bmatrix}, P_5 = \begin{bmatrix} 0 & 1 & 0 \\ X & p & X \\ X & X & X \end{bmatrix}, P_6 = \begin{bmatrix} 0 & X & X \\ 1 & p & X \\ 0 & X & X \end{bmatrix}$$

Figura 1. Casos a cubrir para preservar la conectividad de tipo 4.

Para poder determinar la necesidad de dichas plantillas, se puede notar que la plantilla número P_1 sirve para cubrir las situaciones locales mostradas en la Figura 2.

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & p & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & p & 1 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \\ 1 & p & 1 \\ 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & p & 1 \\ 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \\ 1 & p & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Figura 2. Situaciones que cubre la plantilla P_1 .

Análogamente, la máscara P_2 se usa con los patrones que aparecen en la Figura 3.

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & p & 0 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 0 & p & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & p & 0 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & p & 0 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 0 & p & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Figura 3. Situaciones que cubre la plantilla P_2 .

En todas estas situaciones, el punto p se necesita para preservar la conectividad de tipo 4. Para justificar la necesidad de las máscaras P_3, P_4, P_5, P_6 se pueden considerar los casos ejemplificados en la Figura 4.

$$\begin{bmatrix} y & x & y \\ x & p & x \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} y & x & 0 \\ x & p & 1 \\ y & x & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ x & p & x \\ y & x & y \end{bmatrix}, \begin{bmatrix} 0 & x & y \\ 1 & p & x \\ 0 & x & y \end{bmatrix}$$

Figura 4. Casos donde se emplean las máscaras P_3, P_4, P_5, P_6 .

En este caso, p es necesario para conectar cualquier punto x con los puntos etiquetados como y . Si los puntos x tienen un valor de 1, entonces p es un punto final. Los puntos etiquetados como y no tienen relevancia para la conectividad de tipo 4. Ahora, nos debemos de asegurar que el esqueleto no tenga ramas parásitas como la que se ilustra en la Figura 5.

$$\begin{bmatrix} x & 1 & 1 \\ 0 & p & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figura 5. Caso de rama que debe eliminarse.

En donde p debe ser eliminado independientemente del valor de x . Más aún, existen versiones rotadas del caso de conectividad redundante que se muestra en la Figura 6.

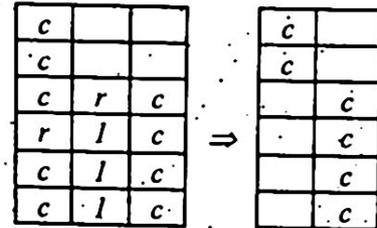


Figura 6. Caso de conectividad redundante.

En la Figura 6, las letras c y r denotan el 4-contorno del objeto. Es claro que solamente uno de los dos puntos etiquetados como r es redundante para el esqueleto, pero usando las seis máscaras de la Figura 1 ambos puntos serán eliminados. Estos problemas puede solucionarse mediante la secuencia expresada para el algoritmos y agregando las máscaras que se muestran en la Figura 7.

$$A_1 = \begin{bmatrix} X & 0 & X \\ 1 & p & x \\ 0 & y & X \end{bmatrix}, A_2 = \begin{bmatrix} 0 & x & X \\ 1 & p & y \\ X & 0 & X \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 1 & X \\ y & p & 0 \\ X & x & X \end{bmatrix}$$

$$A_4 = \begin{bmatrix} X & 1 & 0 \\ 0 & p & x \\ X & y & X \end{bmatrix}, A_5 = \begin{bmatrix} X & y & 0 \\ x & p & 1 \\ X & 0 & X \end{bmatrix}, A_6 = \begin{bmatrix} X & 0 & X \\ y & p & 1 \\ X & x & 0 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} X & x & X \\ 0 & p & y \\ X & 1 & 0 \end{bmatrix}, A_8 = \begin{bmatrix} X & 0 & X \\ y & p & 1 \\ X & x & 0 \end{bmatrix}, \text{ con } x \vee y = 1$$

Figura 7. Máscaras para solucionar la eliminación de ambos puntos de conectividad redundante.

Estas nuevas máscaras eliminan toda redundancia posible en el esqueleto, y se asegura preservar la conectividad de tipo 4, dado que el algoritmo es secuencial. Se pueden combinar las máscaras de la Figura 7 con P_1, P_2 mediante operaciones Booleanas, para obtener las máscaras que aparecen en la Figura 8.

$$B_1 = \begin{bmatrix} a & b & X \\ 1 & p & e \\ c & d & X \end{bmatrix}, B_2 = \begin{bmatrix} c & 1 & a \\ d & p & b \\ X & e & X \end{bmatrix}$$

$$B_3 = \begin{bmatrix} X & d & c \\ e & p & 1 \\ X & b & a \end{bmatrix}, B_4 = \begin{bmatrix} X & e & X \\ b & p & a \\ a & 1 & c \end{bmatrix}$$

donde $c \wedge d = 0, a \wedge b = 0, d + b + e = 1$

Figura 8. Máscaras simplificadas.

Se puede observar que las máscaras P_1 y P_2 son casos particulares de B_1 y B_2 respectivamente, si $c = a = 0$ y $e = 1$. Más aún, si para P_3, P_4, P_5, P_6 los puntos de los lugares opuestos a aquellos etiquetados con una intensidad de 1, tienen un valor de 1, las máscaras de la Figura 9 son casos especiales de B_1, B_2, B_3, B_4 .

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & p & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & p & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & p & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & p & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figura 9. Casos especiales de las máscaras B_1, B_2, B_3, B_4 .

Las cuales indican todas las configuraciones posibles de puntos finales. Por lo que P_3, P_4, P_5, P_6 pueden ser reemplazadas por una sola máscara, la cual aparece en la Figura 10.

$$B_5 = \begin{bmatrix} 0 & h & 0 \\ e & p & g \\ 0 & f & 0 \end{bmatrix}, \text{ con } e + f + g + h = 1$$

Figura 10. Máscara que agrupa los casos de P_3, P_4, P_5, P_6 .

En consecuencia, el conjunto de máscaras:

$$T_4 = \{B_1, B_2, B_3, B_4, B_5\}$$

Es el conjunto que se requiere para el algoritmo de adelgazamiento 4-conectado. Para el caso de las conectividades de tipo 8 y 6, se puede seguir un procedimiento similar para obtener el conjunto de máscaras necesario. En la Figura 11 se muestran

las máscaras para una conectividad de tipo 8 y en la Figura 12 para el caso de conectividad de tipo 6.

$$D_1 = \begin{bmatrix} 1 & 0 & d \\ 0 & p & e \\ h & g & f \end{bmatrix}, \text{ con } d \vee e \vee f \vee g \vee h = 1$$

$$D_2 = \begin{bmatrix} x_1 & e & 0 \\ h & p & f \\ 0 & g & x_2 \end{bmatrix}, \text{ donde } e + f + g + h = 1, x_1 \vee x_2 = 0$$

$$D_3 = \begin{bmatrix} X & 0 & 1 \\ X & p & 0 \\ X & X & X \end{bmatrix}, D_4 = \begin{bmatrix} X & X & X \\ X & p & 0 \\ X & 0 & 1 \end{bmatrix}$$

$$D_5 = \begin{bmatrix} X & X & X \\ 0 & p & X \\ 1 & 0 & X \end{bmatrix}, D_6 = \begin{bmatrix} X & 0 & X \\ 1 & p & 1 \\ X & 0 & X \end{bmatrix}$$

$$D_7 = \begin{bmatrix} X & 1 & X \\ 0 & p & 0 \\ X & 1 & X \end{bmatrix}$$

Figura 11. Máscaras para el algoritmo 8-conectado.

$$E_1 = \begin{bmatrix} X & d & e \\ i & p & f \\ h & g & X \end{bmatrix}, \text{ con } d + e + f + g + h + i = 1$$

$$E_2 = \begin{bmatrix} X & 1 & y_1 \\ 0 & p & y_2 \\ x & 1 & X \end{bmatrix}, \text{ donde } y_1 \wedge y_2 = 0$$

$$E_3 = \begin{bmatrix} X & 0 & x \\ 1 & p & 1 \\ y_1 & y_2 & X \end{bmatrix}, \text{ donde } y_1 \wedge y_2 = 0$$

$$E_4 = \begin{bmatrix} X & x & x \\ 0 & p & x \\ 1 & 0 & X \end{bmatrix}, E_5 = \begin{bmatrix} X & 0 & 1 \\ x & p & 0 \\ x & x & X \end{bmatrix}$$

$$E_6 = \begin{bmatrix} X & x & 0 \\ x & p & 1 \\ x & 0 & X \end{bmatrix}, E_7 = \begin{bmatrix} X & x & x \\ x & p & 0 \\ 0 & 1 & X \end{bmatrix}$$

Figura 12. Máscaras para el algoritmo 6R-conectado.

IV. RESULTADOS.

Los algoritmos fueron implementados usando Entorno Integrado de Desarrollo (IDE) C++ Builder de Borland versión 5.0. En las Figuras 13 y 14 muestran la salida de los algoritmos 4 y conectados con diversos tipos de imágenes.

puede observar que los objetivos perseguidos por los algoritmos fueron cumplidos.

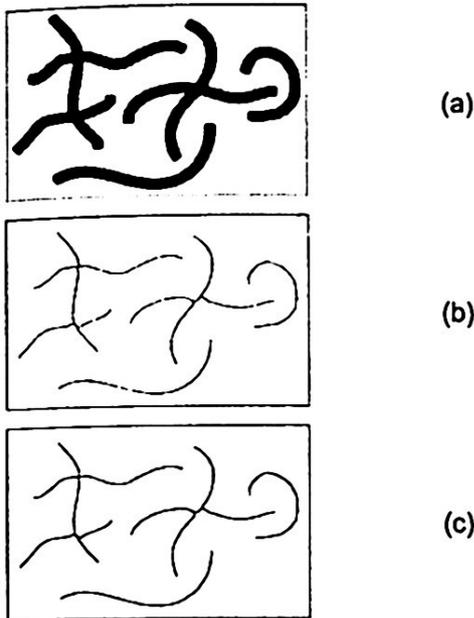


Figura 13. Salida del algoritmo. (a) Imagen de entrada, (b) Esqueleto 8 - conectado, (c) Esqueleto 4 - conectado.

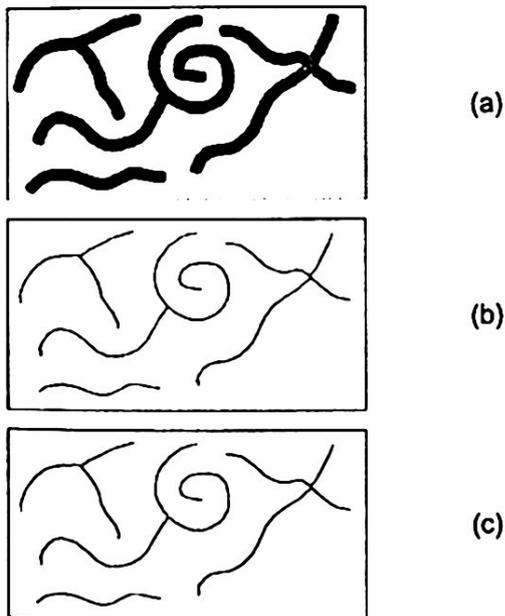


Figura 14. Salida del algoritmo. (a) Imagen de entrada, (b) Esqueleto 8 - conectado, (c) Esqueleto 4 - conectado.

Los objetos ilustrados en las imágenes anteriores, fueron generados manualmente por medio de una herramienta comercial de diseño. Para poder comprobar el funcionamiento del sistema en objetos reales se puede consultar los resultados que aparecen en la Figuras 15 y 16. En algunos casos se presentó la generación de ramas parásitas, para

eliminarlas, se puede aplicar un algoritmo clásico de podado para remover dichas ramas parásitas [13].

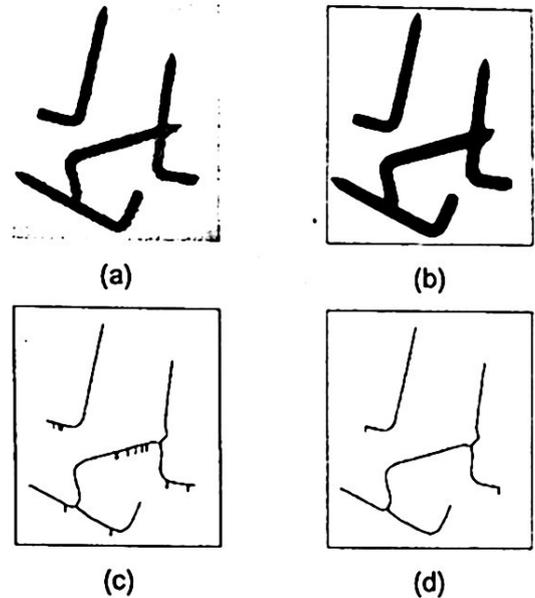


Figura 15. Salida del algoritmo. (a) Imagen de entrada, (b) Imagen binarizada manualmente, (c) Esqueleto 8-conectado sin podar, (d) Esqueleto 8-conectado podado.

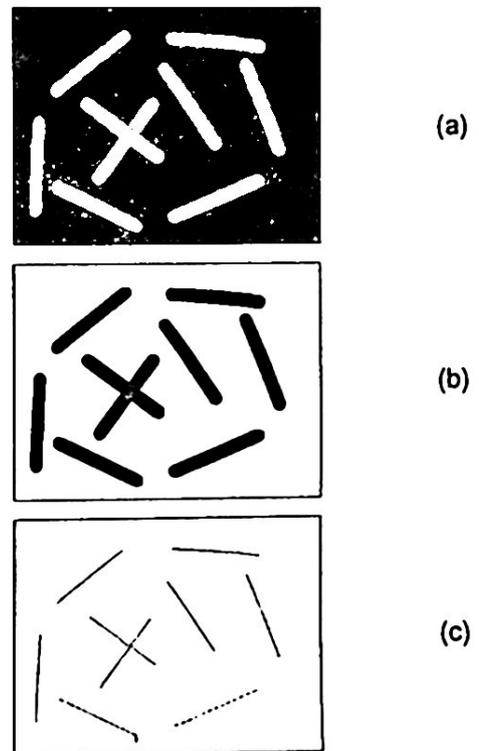


Figura 16. Salida del algoritmo. (a) Imagen de entrada, (b) Imagen binarizada manualmente, (c) Esqueleto 8-conectado sin podar.

CONCLUSIONES.

En este artículo se presentó el esquema general de algoritmo de adelgazamiento que permite obtener esqueletos k-conectados de imágenes digitales binarias. Dicho algoritmo se basa en el empleo de una condición de puntos finales, así como de máscaras o plantillas, que permiten evaluar determinar rápidamente cuando un punto debe o no eliminarse y asegurar la conectividad del esqueleto resultante. De igual manera, las máscaras permiten eliminar los casos de píxeles redundantes. Estos algoritmos pueden ser usados en técnicas que requieran de esqueletos conexos en los que sea de vital importancia preservar las propiedades topológicas de los objetos originales, ya que el tiempo de procesamiento al manipular esqueletos de objetos se ve notoriamente reducido si en lugar de ello, se trabajara con los objetos digitales originales.

De las pruebas realizadas, se pudo comprobar que tiempo de procesamiento fue muy pequeño y en mayoría de las imágenes los resultados fueron correctos y se preservaban las propiedades topológicas de los objetos originales. En la actualidad, se ha trabajado extensamente en el desarrollo de algoritmos de esqueletizado, pero desafortunadamente, muchos de ellos funcionan correctamente con objetos muy simples o con símbolos de escritura de idiomas como el Chino y el Japonés, lo cual limita ampliamente su uso en aplicaciones de propósito general.

Los autores de este trabajo desean extender su agradecimiento al Centro de Investigación en Computación (CIC), a la Coordinación General de Posgrado e Investigación (CGPI) y al Instituto Politécnico Nacional (IPN) el apoyo y las facilidades brindadas para el desarrollo de la presente investigación, así como también a los revisores por sus oportunos y acertados comentarios.

REFERENCIAS.

- [1] A. Rosenfeld, "Characterization of parallel thinning algorithms", *Information and Control*, vol. 29, pp. 286 - 291, 1975.
- [2] T. Pavlidis, "Algorithms for Graphics and Image Processing", Ed. Computer Science Press, 1982.
- [3] J. J. Zou, H.-H. Chang, H. Yan, "Shape skeletonization by identifying discrete local symmetries", *Pattern Recognition*, vol. 34 pp. 185 - 1905, 2001.
- [4] R. W. Smith, "Computer processing of line images: a survey", *Pattern Recognition*, vol. 20, pp. 7 - 15, 1987.
- [5] L. Lam, S.-W. Lee, C. Y. Suen, "Thinning methodologies - a comprehensive survey", *IEEE Tran. Pattern A.*, vol. 14, pp. 869 - 885, 1992.
- [6] F. Leymarie, M. D. Levine, "Simulating the grassfire transform using an active contour model", *IEEE Trans. Pattern A.*, vol. 14, pp. 56 - 75, 1992.
- [7] H. Tamura, "A comparison of line thinning algorithms from the digital viewpoint", *Proc. of 4th ICPR*, pp. 715 - 719, 1987.

- [8] E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal, and triangular arrays", *Communications of the ACM*, vol. 15, pp. 827 - 837, 1972.
- [9] T. Y. Zhang, C.Y. Suen, "A fast parallel algorithm thinning digital patterns", *Image Processing and Computer Vision*, vol. 27, pp. 236 - 239, 1984.
- [10] J. H. Sossa, "An improved parallel algorithm for thinning digital patterns", *Pattern Recognition Letters*, vol. 10, pp. 77 - 80, 1989.
- [11] H. E. Lü, S. P. Wang, "A comment on a fast parallel algorithm for thinning digital patterns", *Image Processing Computer Vision*, vol. 29, pp. 239 - 242, 1986.
- [12] W.H. Abdulla, A. O. Saleh, A. H. Morad, "A preprocessing algorithm for hand written character recognition", *Pattern Recognition Letters*, vol. 7, pp. 13 - 18, 1988.
- [13] R. González, E. Woods, "Digital Image Processing", Second Edition, Prentice Hall.