

ALGORITMO PARA DETECCIÓN DE BORDES Y ULTERIOR DETERMINACIÓN DE OBJETOS EN IMÁGENES DIGITALES

Ledda I. Larcher ^a, Enrique M. Biasoni ^a, Carlos A. Cattaneo ^a, Ana I. Ruggeri ^b, A.
Cecilia Herrera ^a

^aFacultad de Agronomía y Agroindustrias, Universidad Nacional de Santiago del Estero, Av.
Belgrano Sud 1927, G4200ABT Santiago del Estero, Argentina, llarcher@unse.edu.ar,
<http://faa.unse.edu.ar>

^bFacultad de Ciencias Exactas y Tecnología, Universidad Nacional de Santiago del Estero, Av.
Belgrano Sud 1927, G4200ABT Santiago del Estero, Argentina, <http://fce.unse.edu.ar>

Palabras clave: procesamiento de imágenes, segmentación

Resumen. Utilizando el concepto de conectividad se elaboró un algoritmo para reconocer objetos en una imagen digital binaria.

El algoritmo desarrollado inspecciona una imagen píxel a píxel, examinando aquellos que no hayan sido asignados a ningún objeto. Se realiza un recorrido secuencial de la matriz imagen y, para cada píxel con valor 1, se agregan sus coordenadas a una lista. Usando el concepto de 8-vecindad, se inspeccionarán los vecinos buscando aquellos con valor 1 para agregarlos a la lista, al terminar de verificar los píxeles conectados al originalmente encontrado, se han almacenado las coordenadas de cada uno de los píxeles que forman un objeto. De esta manera, cada elemento de la lista es un objeto. El programa desarrollado resultó robusto, demostrando alta eficiencia en distintas aplicaciones, siendo el tiempo computacional directamente proporcional al tamaño de los objetos y con bajo consumo de memoria. Actualmente se trabaja en la relación área perímetro para evitar contar como único dos objetos solapados.

1 INTRODUCCIÓN

La visión artificial tiene como finalidad la extracción de información del mundo físico a partir de imágenes, utilizando para ello una computadora.

Un sistema de visión artificial actúa sobre una representación de una realidad que le proporciona información sobre brillo, colores, formas, etcétera. Estas representaciones suelen estar en forma de imágenes estáticas, escenas tridimensionales o imágenes en movimiento. Se trabajará sobre imágenes estáticas.

La digitalización, es el proceso de paso del mundo continuo (o analógico) al mundo discreto (o digital) mediante el cual se genera una imagen bidimensional, que es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa (por ejemplo su brillo o su matiz).

2 ETAPAS DE UN SISTEMA DE VISIÓN ARTIFICIAL

Se sabe que el ser humano captura la luz a través de los ojos, y que esta información circula a través del nervio óptico hasta el cerebro, donde se procesa. Existen razones para creer que el primer paso de este procesado consiste en encontrar elementos más simples en los que descomponer la imagen (como segmentos y arcos). Después el cerebro interpreta la escena y por último actúa en consecuencia. La visión artificial, en un intento de reproducir este comportamiento, define tradicionalmente cuatro fases principales:

- La primera fase, que es puramente sensorial, consiste en la captura o adquisición de las imágenes digitales mediante algún tipo de sensor.
- La segunda etapa consiste en el tratamiento digital de las imágenes, con objeto de facilitar las etapas posteriores. En esta etapa de procesamiento previo es donde, mediante filtros y transformaciones geométricas, se eliminan partes indeseables de la imagen o se realzan las partes de interés.
- La siguiente fase, conocida como segmentación, consiste en aislar los elementos que interesan de una escena para comprenderla.
- Por último se encuentra la etapa de reconocimiento o clasificación, en la que se pretende distinguir los objetos segmentados, gracias al análisis de ciertas características que se establecen previamente para diferenciarlos.

Estas cuatro fases no se siguen siempre de manera secuencial sino que, en ocasiones, suelen retroalimentarse (Figura 1). Así, puede ser necesario volver a la etapa de segmentación si falla la etapa de reconocimiento, o a la de preproceso, o incluso a la de captura.

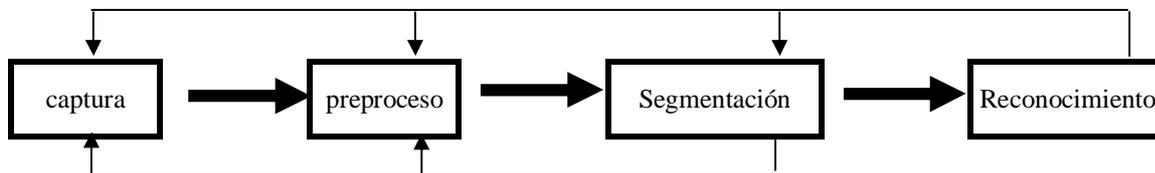


Figura 1. Fases de un sistema de visión artificial

Las imágenes digitales son “señales” discretas, que suelen tener origen en una “señal” continua.

En el proceso de obtención de imágenes digitales se distinguen dos etapas. La primera, conocida como captura, utiliza un dispositivo, generalmente óptico, con el que obtiene información relativa a una escena. En la segunda etapa, que se conoce como digitalización, se transforma esa información, que es una señal con una o varias componentes continuas, en la imagen digital, que es una señal con todas sus componentes discretas.

2.1 Muestreo

La obtención de una imagen digital se realiza mediante un proceso denominado muestreo.

De esta manera se convierte una imagen I_C , que es algo continuo, en una matriz discreta I_D de $N \times M$ píxeles. El número de muestras por unidad de espacio sobre el objeto original lleva al concepto de resolución espacial de la imagen, definida como la distancia entre dos píxeles adyacentes, sobre el objeto original.

De esta forma, el proceso de muestreo para una imagen, que asocia a cada punto un valor real, cambia una imagen del formato (ecuación 1):

$$I_C(x, y) \in R \text{ donde } x, y \in R \quad (1)$$

al formato:

$$I_D(x, y) \in R \text{ donde } x, y \in N \text{ y } 0 \leq x \leq N-1, 0 \leq y \leq M-1$$

que se puede representar en forma matricial (ecuación 2):

$$I_D(x, y) = \begin{pmatrix} I_D(1,0) & I_D(1,1) & \dots & I_D(1,M-1) \\ I_D(2,0) & I_D(2,1) & & I_D(2,M-1) \\ \dots & & & \\ I_D(N-1,0) & I_D(N-1,1) & & I_D(N-1,M-1) \end{pmatrix} \quad (2)$$

3 RELACIONES BÁSICAS ENTRE PÍXELES

Dependiendo de la situación de los píxeles y de los valores que tienen, se definen ciertas relaciones de vecindad y conectividad.

3.1 Vecindad

Para todo punto p de coordenadas (x, y) se dice que un píxel q pertenece a sus 4-vecinos y se escribe $q \in N_4(p)$ si y sólo si q tiene las coordenadas indicadas en la ecuación 3:

$$(x-1, y) \text{ o } (x, y-1) \text{ o } (x+1, y) \text{ o } (x, y+1) \quad (3)$$

Para todo punto p de coordenadas (x, y) se dice que un píxel q pertenece a sus 8-vecinos y se escribe $q \in N_8(p)$ si y sólo si q tiene coordenadas (ecuación 4):

$$(x-1, y) \text{ o } (x, y-1) \text{ o } (x+1, y) \text{ o } (x, y+1) \text{ o } (x-1, y-1) \text{ o } (x-1, y+1) \\ \text{ o } (x+1, y-1) \text{ o } (x+1, y+1) \quad (4)$$

Las figuras 2 y 3 muestran lo enunciado:

	2	
4		5
	7	

Figura 2. Un píxel y sus 4 vecinos

1	2	3
4		5
6	7	8

Figura 3. Un píxel y sus 8 vecinos

3.2 Conectividad

Se ha visto que una imagen se asimila a una matriz, cada uno de cuyos elementos es un píxel. Entre los píxeles de esta matriz se puede definir una relación que define dos píxeles como conectados cuando son vecinos y sus valores son similares desde algún punto de vista. Formalmente (ecuación 5), se define un conjunto V que representa los valores compatibles para que dos píxeles que sean vecinos se diga que están conectados:

$$V = \{ \text{Valores de los píxeles que definen conectividad} \} \quad (5)$$

Se dice que dos píxeles p y q con valores en V están 4-conectados si $q \in N_4(p)$.

Se dice que dos píxeles p y q con valores en V están 8-conectados si $q \in N_8(p)$.

El uso de la 8 conectividad puede dar lugar a ciertas ambigüedades en ciertos análisis de conectividad. Para eliminar esta ambigüedad se define la m-conectividad.

Se dice que dos píxeles p y q con valores en V están m-conectados si $[q \in N_4(p)]$ o $[q \in N_8(p)]$ y $\forall x / x \in (N_4(q) \cap N_4(p))$ x no tiene valores en V . Es decir están m-conectados si están 4 conectados o si están 8 conectados y no tienen ningún 4 vecino común 4 conectado.

3.3 Camino

Un camino desde el píxel p , de coordenadas (x, y) , al píxel q , de coordenadas (s, t) , es una secuencia de píxeles distintos de coordenadas (ecuación 6):

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \quad (6)$$

donde $(x_0, y_0) = (x, y)$ y $(x_n, y_n) = (s, t)$ y (x_i, y_i) está conectado a (x_{i-1}, y_{i-1}) , siendo n la longitud del camino. Se puede hablar de 4, 8 y m-caminos dependiendo del tipo de conexión involucrada.

3.4 Componente conexa

Para todo píxel p de una imagen, el conjunto de los píxeles hasta los que hay un camino desde p se dice que forman su componente conexa. Además se cumple que dos componentes conexas distintas tienen conjuntos de píxeles disjuntos (Vélez Serrano *et al.*, 2003).

4 SEGMENTACIÓN

El fin último de la mayoría de las aplicaciones computarizadas es la identificación de objetos únicos en la imagen que sean relevantes para la aplicación específica. La segmentación se refiere al proceso de separar los objetos de interés del fondo de la imagen. Pueden usarse varias técnicas, que varían desde lo más simple (como umbralizado y máscaras) a lo más complejo (como la detección de bordes). La literatura contiene cientos de técnicas de segmentación pero no existe un método único que pueda considerarse óptimo para todo tipo de imágenes (Bovik, 2009) Los métodos de segmentación varían dependiendo de la modalidad de la imagen, el dominio de aplicación y otros factores específicos. Mientras que algunos métodos emplean técnicas de reconocimiento de patrones basadas en intensidad pura, tales como el umbralizado seguido del análisis de componentes conectados (Higgins *et al.* 1989, Niki *et al.* 1993), otros métodos aplican modelos específicos para extraer información (Molina *et al.* 1998, Klein *et al.*, 1997). Dependiendo de la calidad de la imagen y de factores generales como -por ejemplo- el ruido, algunos métodos de segmentación podrían requerir un pre procesamiento de la imagen antes de aplicar el algoritmo de segmentación (Guo y Richardson, 1998, Sato *et al.*, 1998). Por otra parte, algunos métodos aplican post

procesamiento para resolver problemas que surgieran de una sobre-segmentación. Sobre todo, los métodos de segmentación pueden ser ampliamente categorizados en basados en un punto, basados en bordes y basados en regiones (Bovik, 2009)

La segmentación es el proceso que divide una imagen digital en regiones homogéneas con respecto a una o más características (como, por ejemplo, el brillo o el color) con el fin de facilitar un posterior análisis o reconocimiento automático. Localizar la cara de una persona dentro de la imagen de una fotografía o encontrar los límites de una palabra dentro de una imagen de un texto, son ejemplos de problemas de segmentación.

De acuerdo a Baxes (1994), la segmentación debe verse como un proceso que, a partir de una imagen, produce otra en la que cada píxel tiene asociada una etiqueta distintiva del objeto al que pertenece. Así, una vez segmentada una imagen, se podría formar una lista de objetos consistentes en las agrupaciones de los píxeles que tengan la misma etiqueta.

La segmentación termina cuando los objetos extraídos de la imagen se corresponden unívocamente con las distintas regiones disjuntas a localizar en la misma. En este caso se habla de segmentación completa de la escena o imagen y en el caso contrario, de segmentación parcial. En una escena compleja, el resultado de la segmentación podría ser un conjunto de regiones homogéneas superpuestas y en este caso, la imagen parcialmente segmentada deberá ser sometida después a un tratamiento posterior con el fin de conseguir una segmentación completa.

El proceso de segmentación de una imagen depende del problema que se desee resolver. Por ejemplo, sobre una imagen de una página de texto se pueden segmentar las palabras y los caracteres que las forman (si se desea hacer OCR de los mismos), o los logotipos y membretes (si se desea clasificar el documento), etc. Por ello, dentro de una misma imagen pueden realizarse diferentes segmentaciones.

5 ALGORITMO

Utilizando el concepto de componente conexas, se puede plantear el detectar los objetos presentes en una imagen al encontrar las componentes conexas de la misma. Esto ocurre cuando los objetos tienen un color uniforme y distinto del fondo, lo que permite asegurar que los contornos del objeto se corresponden con los bordes de la componente conexas. En el caso de imágenes en blanco y negro las componentes conexas suelen corresponder a los objetos directamente.

Se usa una matriz O , cuyo tamaño cambiará a lo largo de la ejecución del programa. En ella se almacenarán las coordenadas de los píxeles que componen a los diferentes objetos.

Se toma una imagen digital binarizada, representada en una matriz I de $N \times M$ píxeles, y se recorre de izquierda a derecha, de arriba hacia abajo.

La inspección de vecinos se realiza tomando como casos especiales las esquinas superiores derecha e izquierda, los bordes izquierdo y derecho, las esquinas inferiores izquierda y derecha y el borde inferior.

Se recorre la matriz I elemento a elemento.

Para cada píxel $p(i, j) = 1$ se anotan sus coordenadas en una lista y se buscan entre sus 8 vecinos aquellos que estén conectados.

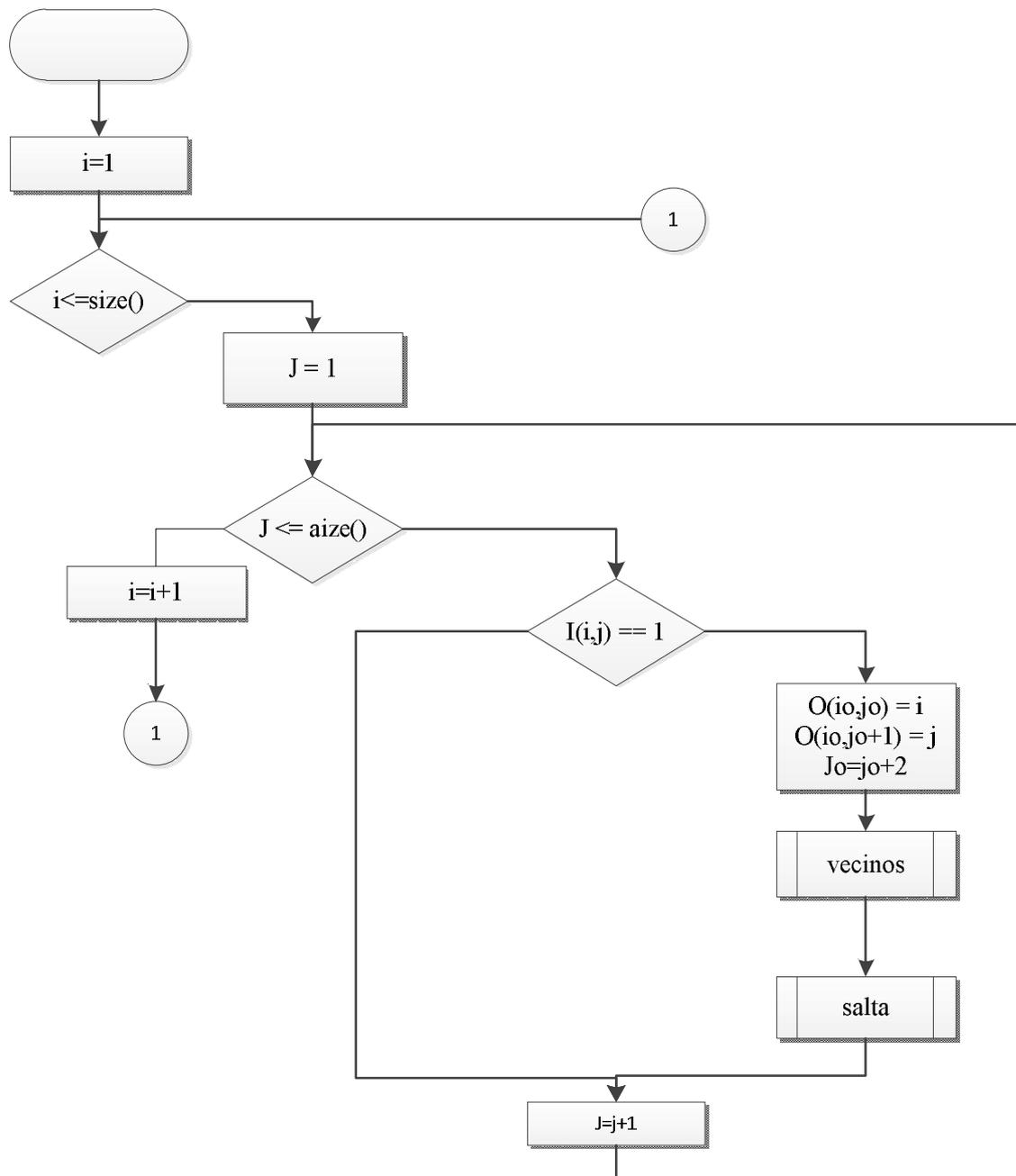
Cada vecino $v(k, l)$ que se encuentra con el valor 1, es agregado a la matriz O , a continuación del píxel $p(i, j)$. En este momento, el interés se centrará en el píxel $v(k, l)$ para quien se buscarán los vecinos distintos de cero. A medida que se los encuentre, se chequea que no hayan sido previamente seleccionados y se los agrega a la matriz O pasando a ser el nuevo píxel de interés.

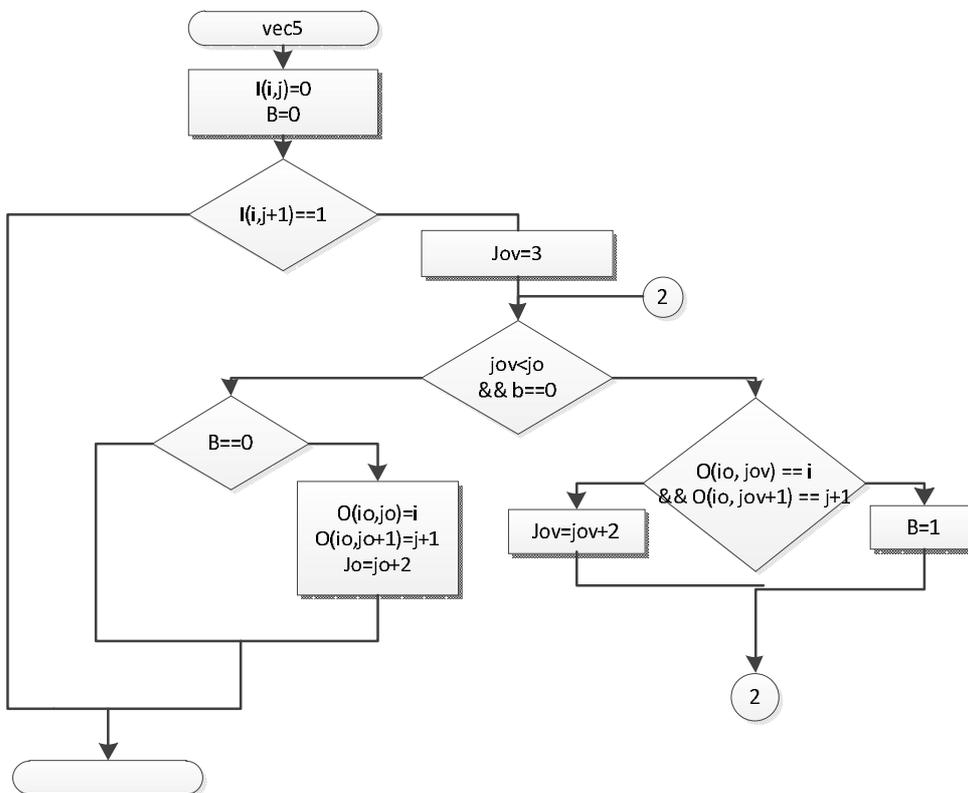
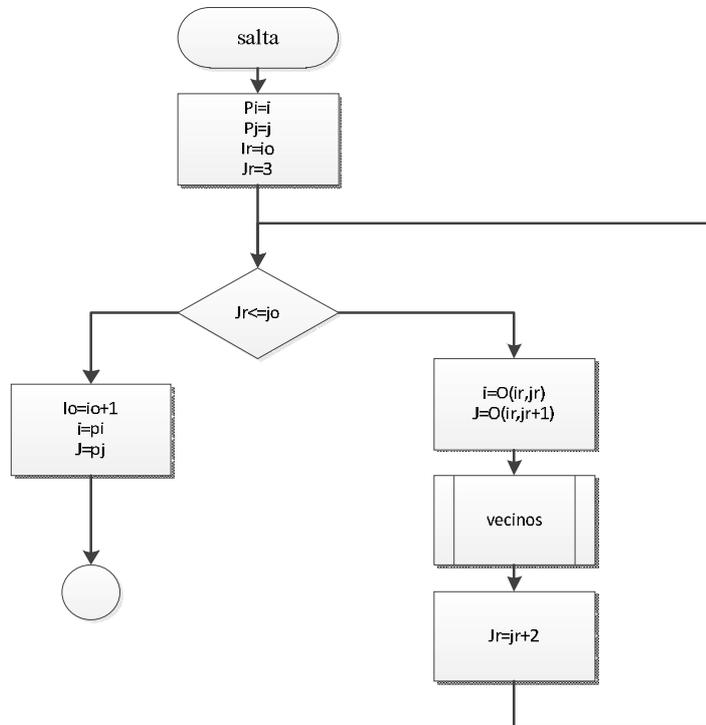
A efectos de no agregar píxeles repetidos (con el consecuente consumo de memoria) ni inspeccionar píxeles ya chequeados (lo que insumiría tiempo computacional), cada uno es anulado en el momento de agregarlo a la lista.

El proceso se repite hasta no encontrar más vecinos conectados. En este momento se incrementa el índice fila de la matriz O que, de esta manera, almacena un trayecto por fila, cada uno de éstos será considera un objeto de la imagen.

El algoritmo retorna el control al recorrido general de la matriz, pasando $p(i, j+1)$ a ser el nuevo píxel de interés y se continúa el recorrido secuencial sobre I .

La figura 4 muestra el diagrama de bloques del proceso descrito y puede observarse en la figura 5 un ejemplo práctico





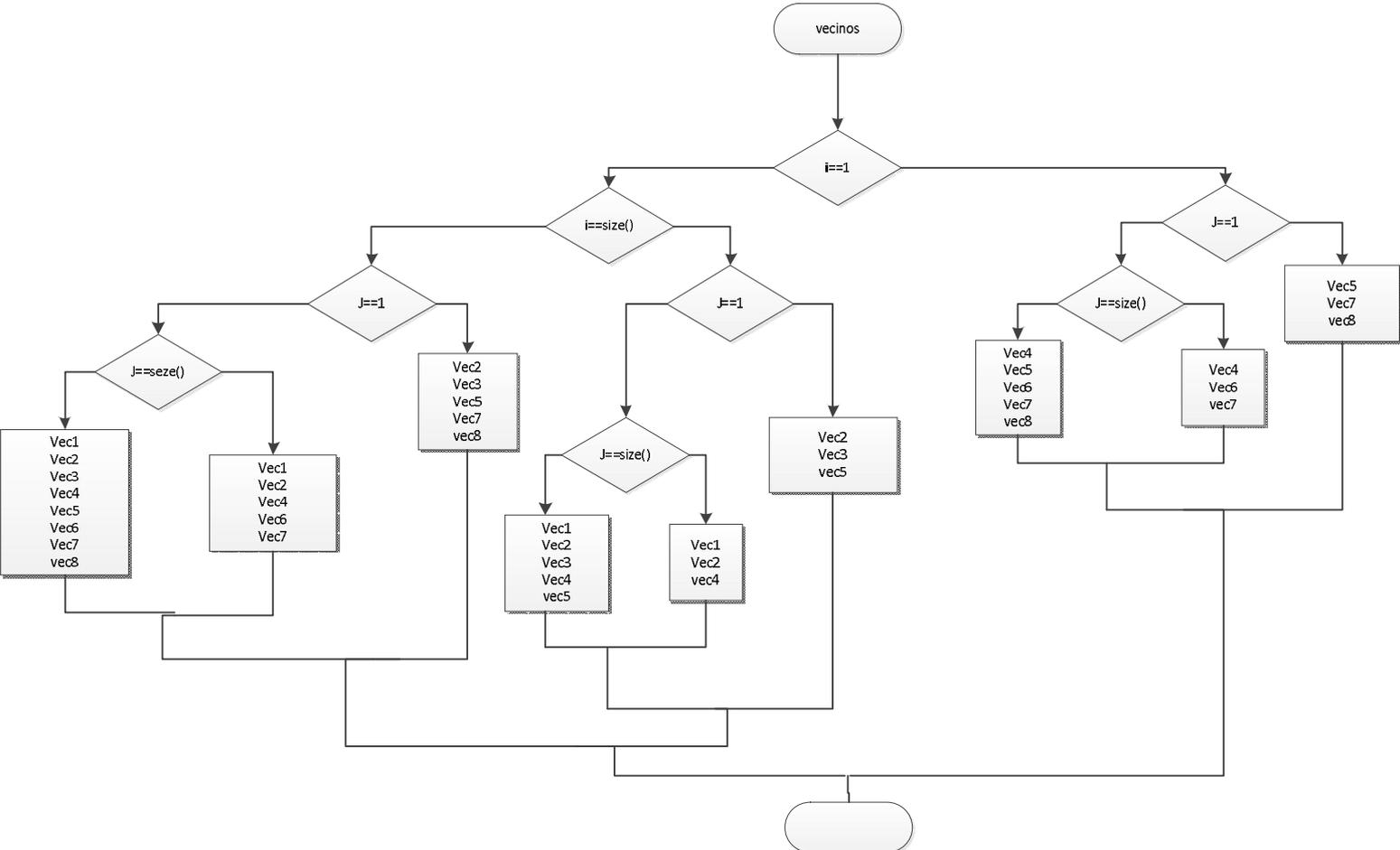


Figura 4. Diagramas de los diferentes procesos. Sólo se presenta la rutina vec5 ya que sólo deben cambiarse los índices

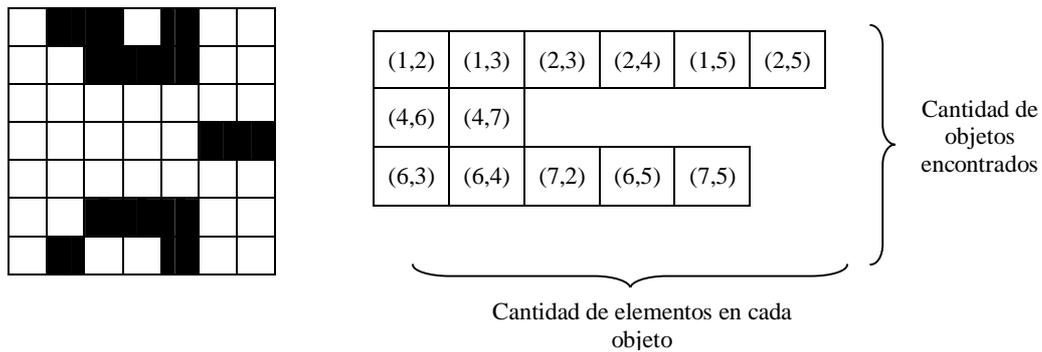


Figura 5. Matriz I con la lista de objetos encontrados. Por cada objeto se listan las coordenadas de cada elemento

6 RESULTADOS

El algoritmo presentado se programó usando Matlab utilizando una PC con procesador Intel Core2 Quad Q8200 con 2 Gigabytes de RAM.

Para realizar el conteo de objetos se utilizaron imágenes color, que fueron sometidas a un pre-procesamiento usando algoritmos de umbralizado por entropías de Shannon y el método de Otsu con el objetivo de obtener imágenes binarizadas.

Las imágenes en formato jpg se transforman a matrices para su procesamiento.

A partir de imágenes generadas por simulación de crecimiento de granos (Cattaneo y Silveti, 2009), se obtuvo la figura 6, que representa un crecimiento de granos en aluminio a 773 K con 20 orientaciones cristalográficas distintas en un tiempo de simulación de 480 lazos. En la Figura 7 puede observarse el resultado del pre procesamiento. Por último, la Figura 8 se muestra el conteo manual, que coincide con el número de granos encontrados por el programa (44 objetos). La imagen es de 300 x 300 y el tiempo de procesamiento fue de 31 minutos

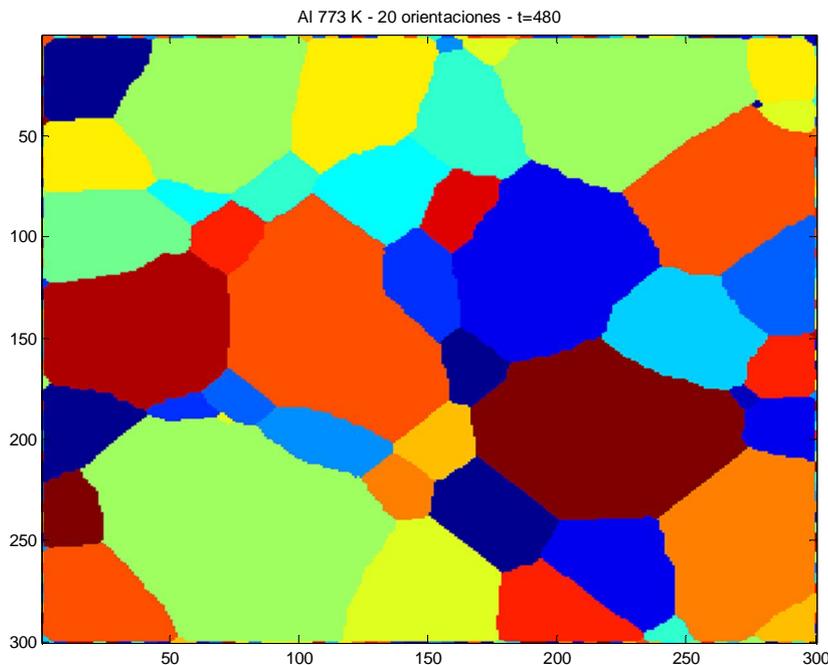


Figura 6. Crecimiento de granos en aluminio a 773 K con 20 orientaciones cristalográficas distintas en un tiempo de simulación de 480 lazos

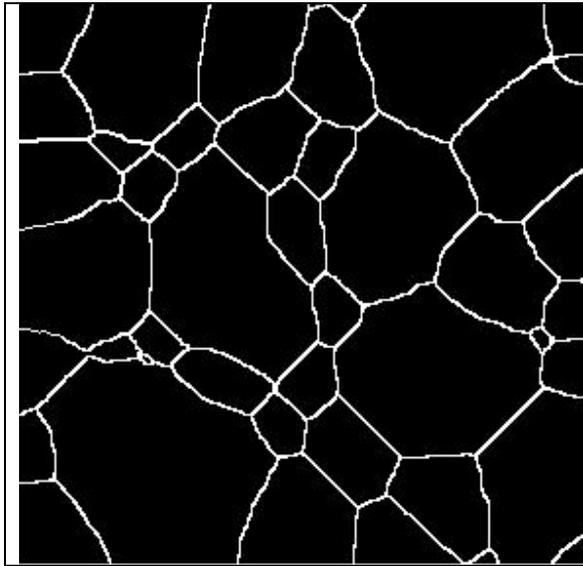


Figura 7. Imagen binarizada

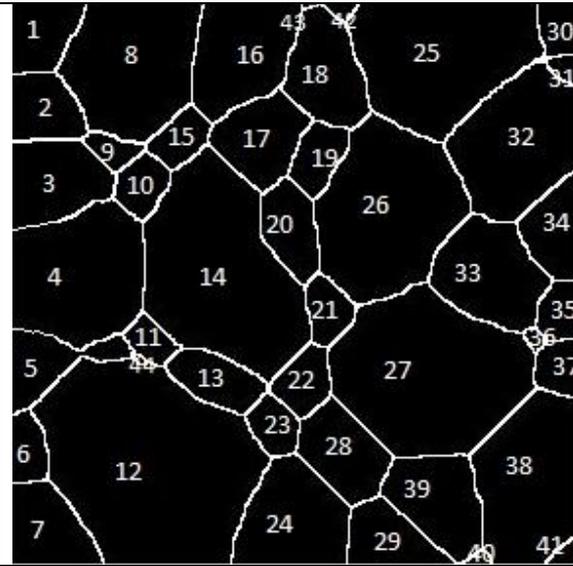


Figura 8. Conteo manual de objetos

A los fines de probar la eficacia se utilizó la [Figura 9](#) que tiene como dimensiones 440x231 píxeles. Los resultados entre el conteo usando el programa y el realizado manualmente coincidieron (12 objetos) y fue realizado en 173 segundos.



Figura 9

A continuación se probó utilizando la [Figura 10](#) en la que, no sólo varía la cantidad de objetos, sino también el tamaño de los mismos. Los resultados obtenidos no coincidieron con el conteo manual, aunque una verificación posterior permitió determinar que se debía a defectos en la conexión de los píxeles, como se muestra en la [Figura 11](#). El tiempo de procesamiento fue de 442 segundos para una imagen de 450x470 píxeles.

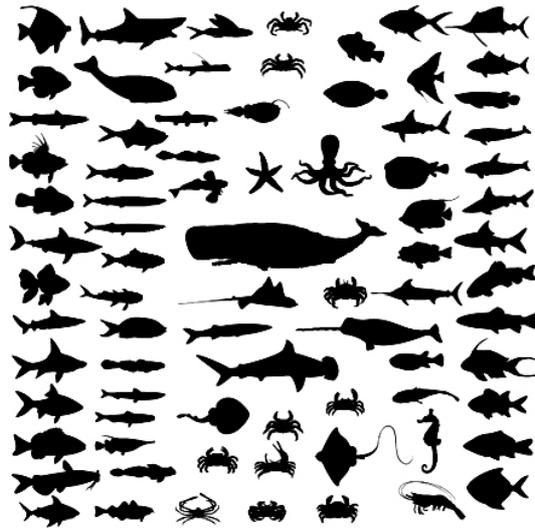


Figura 10

7 CONCLUSIONES

El programa demostró alta eficiencia en distintas aplicaciones, siendo el tiempo computacional directamente proporcional al tamaño de los objetos.

El proceso es altamente dependiente de una correcta umbralización como pudo comprobarse al usar la figura 10 sin realizar umbralizado. En la [Figura 11](#) se muestra la distorsión y diferentes tonos de gris para que el ojo humano perciba al grupo de píxeles como una curva; durante el procesamiento, los niveles de gris muy claros eran tomados como blanco, los oscuros como negro resultando así pequeños objetos separados, por lo que el conteo era incorrecto.

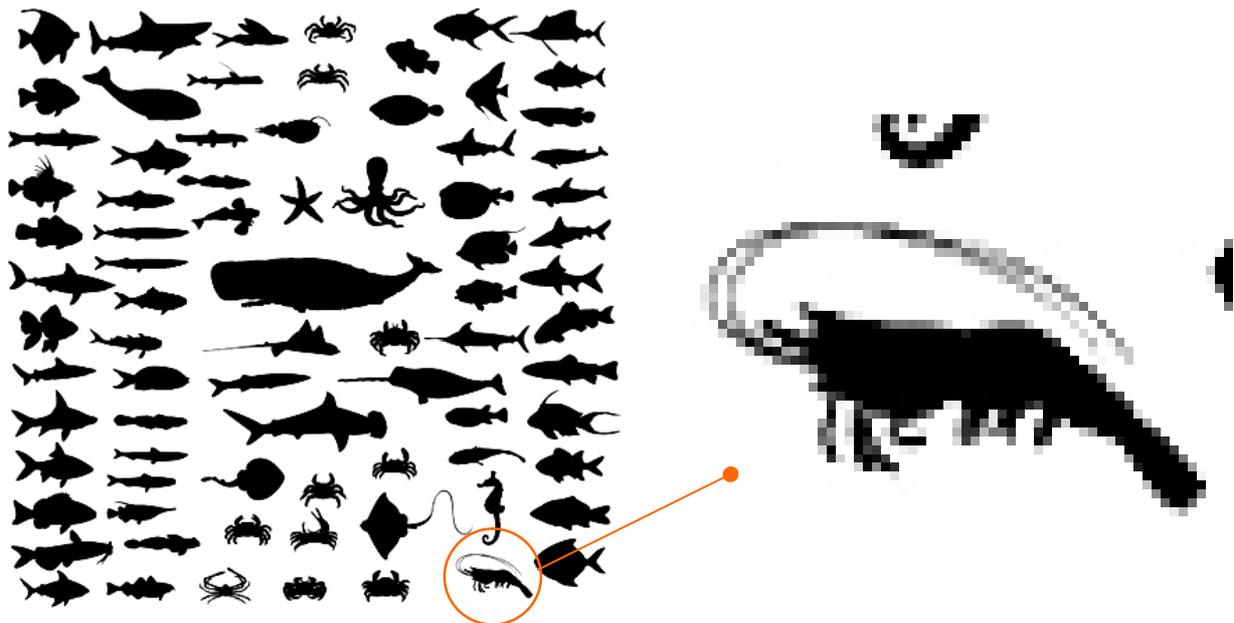


Figura 11. Imagen sin umbralizar y detalle de uno de los objetos

Al probar el funcionamiento del programa, con una matriz cuadrada con la diagonal principal de valor 1 (uno) el resultado es correcto; sin embargo si la situación fuera al

contrario (que la diagonal principal se encuentre en cero dividiendo la matriz en dos objetos), el conteo obtenido es incorrecto debido a la comprobación del vecino 6, con lo que se contabiliza como si fuera un objeto único. (Ver Figura 12). Sin embargo, situaciones de este tipo (objetos con sólo un píxel de separación entre los mismos) son muy poco probables en imágenes reales.

El equipo de investigación se encuentra en este momento trabajando en la resolución del problema antes mencionado, así como en la relación área perímetro para evitar contar como único dos objetos solapados.

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

Figura 12. Situación problemática al comprobar los vecinos

REFERENCIAS

- Baxes, G.A., *Digital Image Processing: Principles and Applications*, J. Wiley & Sons, 1994.
- Bovik A., *The essential guide to image processing*. Academic Press, 2009.
- Cattaneo, C. A., Silveti, S. P., Efectos de la difusión en borde de grano en el crecimiento anormal de grano. *Mecánica Computacional XXVIII*: 2425-2433, 2009.
- Guo, D. and Richardson, P., Automatic vessel extraction from angiogram images. *IEEE Comput. Cardiol.*, 25:441–444, 1998.
- Higgins, W. E., Spyra, W. J. T., Ritman, E. L., Kim, Y., and Spelman, F. A., Automatic extraction of the arterial tree from 3-D angiograms. *IEEE Conf. Eng. Med. Biol.*, 2:563–564, 1989.
- Klein, A. K., Lee, F., and Amini, A. A., Quantitative coronary angiography with deformable spline models. *IEEE Trans. Med. Imaging*, 16:468–482, 1997.
- Molina, C., Prause, G., Radeva, P., and Sonka M., 3-D catheter path reconstruction from biplane angiograms. *SPIE*, 3338:504–512, 1998.
- Niki, N., Kawata, Y., Satoh, H., and Kumazaki T., 3D imaging of blood vessels using x-ray rotational angiographic system. *IEEE Med. Imaging Conf.*, 3:1873–1877, 1993.
- Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., and Kikinis, R., 3Dmultiscale line filter for segmentation and visualization of curvilinear structures in medical images. *IEEE Med. Image Anal.*, 2:143–168, 1998.
- Vélez Serrano, J. F., Moreno Díaz, A. B., Sánchez Calle, Á., Esteban Sánchez-Marín J. L., *Visión por computador*. Editorial Dykinson, 2003.