

# Implementación de un algoritmo para la detección y conteo de células en imágenes microscópicas

Moreira Quiroz, José <sup>(1)</sup>, Valencia Delgado, Vladimir <sup>(2)</sup>, Chávez Burbano, Patricia <sup>(\*)</sup>

Facultad de Ingeniería en Electricidad y Computación (FIEC)

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, Km 30.5 vía Perimetral

Apartado 09-01-5863. Guayaquil, Ecuador <sup>(1)(2)(\*)</sup>

jmoreira@fiec.espol.edu.ec <sup>(1)</sup>, pvalenci@fiec.espol.edu.ec <sup>(2)</sup>

(\*) Chávez Burbano, Patricia, MSc, ESPOL, pchavez@fiec.espol.edu.ec

## Resumen

*En el presente trabajo, presentamos el desarrollo de una aplicación para el conteo de células en imágenes obtenidas mediante microscopio, en la cual se hace posible la utilización de 2 algoritmos (métodos) de segmentación y de 2 de los principales operadores para detectar bordes en imágenes.*

*En esta aplicación, hemos trabajado con una serie de imágenes de células obtenidas en análisis de diferentes fluidos y tejidos, encontradas en su mayoría en la Internet, que para propósitos de demostrar el desempeño de nuestra aplicación, son aceptables.*

*Para la realización de la misma, hemos empleado el muy conocido paquete computacional Matlab 7.7.0 (R2008b), de la compañía MathWorks, el que dispone de una serie de funciones que facilitan el procesamiento de imágenes.*

**Palabras Claves:** Algoritmo, conteo, células, segmentación, operadores.

## Abstract

*In this work, we develop an application for counting cells in images obtained through microscope, in which is possible the use of 2 algorithms (methods) of segmentation and 2 of the main operators to detect edges in images.*

*In this application, we have worked with several images of cells obtained in analysis of different fluids and tissues, found in their majority on Internet, to porpoise of show the performance of our application, are acceptable.*

*To develop our application, we have used the much known computational packet Matlab 7.7.0 (R2008b), of the MathWorks company, which have some functions that become easy the image processing.*

**Key Words:** Algorithm, count, cells, segmentation, operators.

## 1. Introducción

En las ciencias médicas, existen varios tipos de exámenes, como el conteo de células T, conteo de glóbulos rojos o el Frotis de sangre que requieren determinar la cantidad de células presentes en una muestra, esto con el fin de descartar o confirmar la presencia de alguna enfermedad o cuando el médico sospecha de una anomalía de algún tipo de célula.

Uno de los métodos que se emplea en la actualidad se denomina el de la Cámara de conteo celular o Cámara de Neubauer.

La exactitud y la velocidad con que se obtiene el resultado de este examen dependen en parte, de la experiencia de la persona que examina la muestra, ya que el conteo se realiza de forma manual.

En la búsqueda de una mejora de este tipo de procedimiento nos hemos propuesto desarrollar una forma diferente de conteo de células a través de un programa que permita acelerar y aportar un mayor grado de exactitud al proceso de conteo de células utilizando para esto imágenes de células obtenidas a través de un microscopio.

Para ello, nuestro trabajo se basará en técnicas de segmentación y operadores para la detección de bordes, los cuales analizaremos para encontrar los que mejor se adapten a nuestro requerimiento.

## 2. Objetivos

- Implementar un algoritmo que permita contabilizar células en imágenes microscópicas.

- Lograr que el conteo de células que se realiza con el algoritmo sea más rápido con este, que con los métodos tradicionales.
- Lograr un mayor grado de exactitud en el conteo.
- Desarrollar una aplicación que permita ejecutar este algoritmo.

### 3. Conceptos Generales

El procesamiento digital de señales nos permite extraer información considerada relevante, o modificarla para darle un uso apropiado, empleando para ello herramientas computacionales.

Se puede emplear el procesamiento digital de señales en una gran variedad de aplicaciones tales como: transmisión de información a través de canales de comunicación, restauración y mejoramiento de imágenes y de audio, reconocimiento de objetos en imágenes, medición de parámetros tales como la velocidad en objetos en desplazamiento capturados en video, etc.

Nuestro proyecto empleará la detección de objetos basada a su vez en el proceso conocido como Segmentación de Imágenes.

#### 3.1 Definición de Imagen Digital

Una imagen digital, puede ser definida como una función en dos dimensiones,  $f(x,y)$ , donde:  $x$  y  $y$  son coordenadas espaciales, la amplitud de  $f$  en el punto  $(x,y)$  es conocida como la intensidad de la imagen en ese punto. Tanto, los valores de las coordenadas  $x$  y  $y$ , como de la amplitud de la función  $f$  son valores finitos.

El término nivel de gris, se lo emplea para referirse a la intensidad en imágenes monocromáticas.

Las imágenes a color, están formadas por una combinación de imágenes individuales en dos dimensiones.

Una imagen está compuesta por un número finito de elementos (puntos) con una ubicación y valor de intensidad particular, donde cada elemento es conocido como pixel.

#### 3.2 Transformación de Intensidad

Una transformación espacial opera directamente sobre los pixeles de una imagen. La transformación de intensidad en el dominio espacial, es denotada por la expresión:

$$g(x, y) = T[f(x,y)] \quad (3.1)$$

Donde  $f(x,y)$  es la imagen sobre la cual se aplicará la transformación,  $g(x,y)$  es la imagen procesada y  $T$  es un operador sobre  $f$ , definido sobre un vecindario especificado cerca al punto  $(x,y)$ .  $T$  puede operar sobre un conjunto de imágenes, por ejemplo sumando  $K$  imágenes para provocar una reducción de ruido.

### 3.3 Filtrado espacial

El procedimiento que se lleva a cabo al realizar el filtrado espacial es el que sigue:

- Se escoge un punto central,  $(x,y)$ .
- Se desarrolla una operación que involucre solo los pixeles que rodean al punto central.
- El resultado de esa operación será la "respuesta" del proceso (filtrado) en ese punto.
- El proceso es repetido para cada punto en la imagen.

A medida que el punto central es movido, se crean nuevos vecindarios, (siendo los vecindarios, el conjunto de pixeles que rodean al punto central incluido dicho punto), uno para cada pixel en la imagen  $f$ . Si los cálculos realizados sobre los pixeles de un vecindario son lineales, la operación es llamada filtrado espacial lineal, de otra manera es llamada filtrado espacial no lineal. El filtrado espacial también es conocido como procesamiento de vecindario.

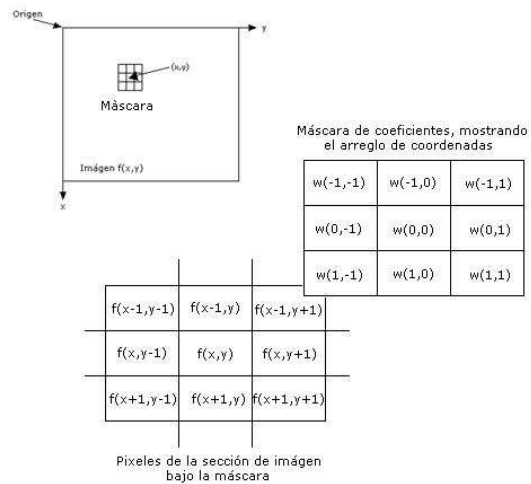


Figura 3.1: Filtrado espacial

Existen dos conceptos muy importantes, relacionados estrechamente e involucrados en el filtrado espacial lineal. Uno es la correlación; el otro es la convolución. La Correlación es el proceso de pasar la máscara  $w$  por el arreglo de imagen  $f$ , en la forma descrita en la figura 1.2. La Convolución es el mismo proceso, excepto que  $w$  es rotado  $180^\circ$  antes de ser pasada sobre  $f$ .

### 3.4 Explicación Gráfica de la Correlación y la Convolución

Para esta explicación emplearemos la figura 3.2. La figura 3.2(a) será la función que representará la imagen que vamos a filtrar. La función  $w(x,y)$  en la figura 3.2(b) será empleada como máscara de filtro en la Correlación o en el caso de realizar la Convolución, la máscara que se utilizará será la representada por la función  $-w(x,y)$  en la figura 3.2(c), que es la versión rotada  $180^\circ$  de la función  $w(x,y)$ .

En la figura 3.2(d) se muestra la función  $f(x,y)$  a la cual se la ha rellenado con dos hileras de ceros en su contorno, esto con el fin de que ninguno de los elementos en la máscara se quede sin un elemento correspondiente en la imagen con el cual trasladarse.

En la figura 3.2(e) se muestra el inicio de la Correlación, donde se observa que esta empieza con el elemento en la parte inferior derecha de la máscara (número 9) trasladado con el elemento en el origen de la función  $f(x,y)$  (número 0).

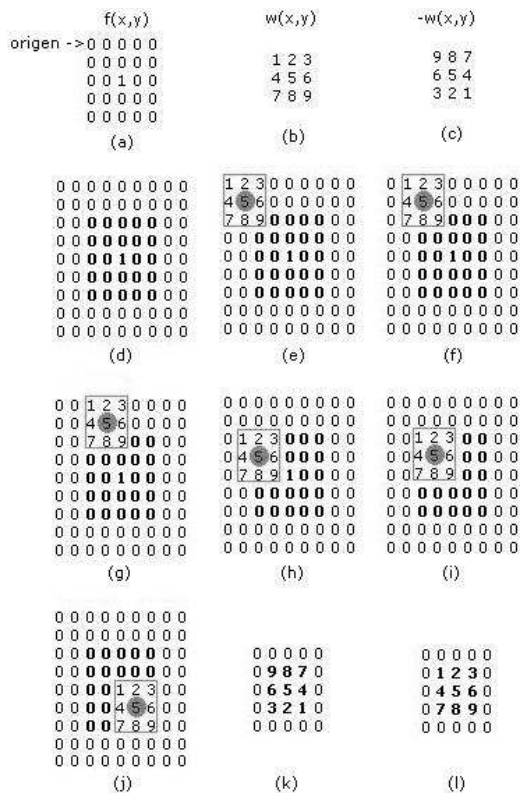


Figura 3.2: Explicación gráfica de la Correlación y Convolución

La ubicación del elemento central de la máscara (número 5), (que se encuentra encerrado en un círculo) será finalmente la ubicación del resultado de la Correlación. Las figuras 3.2(f) y 3.2(g) muestran el desplazamiento que se realiza de la máscara sobre la imagen. La figura 3.2(h)

muestra un instante antes de que la máscara se traslape con el primer elemento diferente de cero en la imagen (el número 1), la figura 3.2(i), cuando se ha producido este traslape y en la figura 3.2(j), se muestra el instante en que el último elemento de la máscara se ha trasladado con el único elemento diferente de cero en la imagen. Finalmente, en la figura 3.2(k) se aprecia el resultado de la correlación desarrollada. En caso de utilizar la máscara de la figura 3.2(c), lo que se llevará a cabo es la convolución, cuyo resultado se puede apreciar en la Figura 3.2(l).

### 3.5 Segmentación de Imágenes

La segmentación, es un proceso en el cual una imagen es sub-dividida en las regiones u objetos que la componen. La segmentación concluye cuando los objetos de interés han sido aislados. En nuestro caso, la segmentación debería de concluir cuando las células de las imágenes, hayan sido aisladas.

Los algoritmos de segmentación para imágenes monocromáticas generalmente están basados ya sea en la discontinuidad o en la similitud de los valores de intensidad en las imágenes.

En la primera categoría, y en la cual estará basado nuestro proyecto, el enfoque es para particionar una imagen basado en cambios abruptos de intensidad, tal como ocurre en los bordes de una imagen. Los algoritmos basados en la detección de bordes, son los más básicos que existen en el tratamiento del tema de segmentación. Los principales algoritmos en la segunda categoría se basan en particionar la imagen en regiones que son similares de acuerdo a un conjunto de características predefinidas.

#### 3.5.1 Detección de Discontinuidades basada en valores de Intensidad

La forma básica para encontrar discontinuidades, es pasar una máscara a través de la imagen, en la forma descrita en la sección 3.3. Para una máscara de  $3 \times 3$  este procedimiento involucra computar la suma de productos de los coeficientes con los niveles de intensidad contenidos en la región abarcada por la máscara. Esto es, la respuesta  $R$  de la máscara en cualquier punto  $(x, y)$  de la imagen está dado por:

$$R = \sum_{k=1}^9 w_k z_k \quad (3.2)$$

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 \quad (3.3)$$

Donde  $z_k$  es la intensidad del pixel asociado con el coeficiente  $w_k$  de la máscara.

### 3.5.2 Detección de Bordes

Un borde, dentro de una imagen, es considerado un tipo de discontinuidad, por lo que este puede ser detectado usando derivadas de primer y segundo orden.

La derivada de primer orden para una imagen es el gradiente, el cual, para una función en dos dimensiones,  $f(x,y)$  viene dado por:

$$\nabla f = [G_x \ G_y] = \left[ \frac{df}{dx} \ \frac{df}{dy} \right] \quad (3.4)$$

La magnitud de este vector es:

$$\begin{aligned} \nabla f &= \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \quad (3.5) \\ &= \left[ \left( \frac{df}{dx} \right)^2 + \left( \frac{df}{dy} \right)^2 \right]^{1/2} \quad (3.6) \end{aligned}$$

Esta cantidad se puede simplificar en algunos casos, omitiendo la raíz cuadrada de la operación:

$$\nabla f = G_x^2 + G_y^2 \quad (3.7)$$

O usando el valor absoluto de los términos:

$$\nabla f = |G_x| + |G_y| \quad (3.8)$$

Esta aproximación, se comporta también como una derivada, esto quiere decir, que se hace cero en áreas de intensidad constante y sus valores son proporcionales al grado de cambio de intensidad, en áreas donde los valores de los pixeles son variables. Es común referirse a la magnitud del gradiente o su aproximación simplemente como “el gradiente”.

### 3.5.3 Detector de bordes Sobel

El detector de borde Sobel usa la máscara que se muestra en la figura 3.3 para aproximar digitalmente las primeras derivadas  $G_x$  y  $G_y$ , de tal manera, que el gradiente en el punto central de un vecindario puede ser computado por el detector Sobel como sigue:

$$[G_x^2 + G_y^2]^{1/2} \quad (3.9)$$

$$= \{ [(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)]^2 + [(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)]^2 \}^{1/2} \quad (3.10)$$

Entonces, un pixel en la ubicación  $(x,y)$  es considerado un pixel de borde si  $g \geq T$  en esa ubicación, donde  $T$  es un umbral que debe ser especificado.

-1	-2	-1
0	0	0
1	2	1

$$G_x = \frac{(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)}{(z_1 + 2z_2 + z_3)}$$

-1	0	1
-2	0	2
-1	0	1

$$G_y = \frac{(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)}{(z_1 + 2z_4 + z_7)}$$

Figura 3.3: Máscaras del detector Sobel

### 3.5.4 Detector de bordes Canny

El detector Canny es considerado, el más efectivo a la hora de detectar bordes, debido a su eficacia. Este detector, realiza su trabajo de la siguiente manera:

1. La imagen se suavizada usando un filtro Gaussiano con una desviación estándar, esto para reducir el ruido. El gradiente local, y la dirección del borde, son computadas en cada punto. El detector Sobel, puede ser usado para computar  $G_x$  y  $G_y$ .

2. Se define un punto de borde, como un punto cuyo peso es localmente máximo en la dirección del gradiente.

3. Los puntos de borde determinados en (2) originan crestas en la dirección de crecimiento del gradiente de la imagen. El algoritmo luego rastrea a lo largo de la cima de estas crestas, y lleva a cero los pixeles que no están en realidad sobre la cima de la cresta, originando una línea delgada en la salida, un proceso conocido como supresión no máxima. Los pixeles de crestas son luego comparados usando dos umbrales,  $T_1$  y  $T_2$  con  $T_1 < T_2$ . Los pixeles de crestas con valores mayores que  $T_2$  se dice que son “probables candidatos” para ser pixeles de borde. Los pixeles en crestas con valores entre  $T_1$  y  $T_2$  se dice que son “candidatos poco probables” para ser pixeles de borde.

4. Por último, el algoritmo realiza la unión de los pixeles, incorporando “candidatos débiles” que están 8-conectados a los pixeles “probables”.

### 3.6 Etiquetado de componentes en imágenes binarias

Una representación común de una imagen digital en dos dimensiones, es el de un arreglo de pixeles asociados con colores en un mapa de color. Un caso particular, es el de las imágenes binarias, las cuales son imágenes digitales que pueden ser representadas usando un mapa de dos colores (Primer Plano y Fondo). En las imágenes binarias, la información está contenida en forma de componentes y en la interrelación entre ellos.

### 3.8 Etiquetado de Segmentos

El resultado de la segmentación satisfactoria de una imagen es el etiquetado de cada pixel correspondiente a un segmento determinado.

El objetivo del etiquetado es asignar a cada pixel de una imagen el número de etiqueta o índice del segmento al que pertenece.

De manera general este procedimiento puede ser descrito de la siguiente manera:

Dada una imagen binaria y el conjunto de pixeles pertenecientes al primer plano  $F$  de la imagen, la meta es asociar una etiqueta de componente  $LABEL(p)$  a cada pixel  $p$  en la imagen, tal que:

1)  $LABEL(p) = 0$  si  $p \in F^c$  y  $LABEL(p) = +\infty$  si  $p \in F$

2)  $LABEL(p) = LABEL(q)$  si y solo si los pixeles  $p \in F$  y  $q \in F$  están en el mismo componente conectado. De otra forma  $LABEL(p) \neq LABEL(q)$ .

El mapa de etiquetas  $(LABEL(p))_{p \in F}$  primero es inicializado usando la siguiente ecuación:

$$\begin{cases} LABEL^{(0)}(p) = +\infty & \forall p \in F \\ LABEL^{(0)}(p) = 0 & \forall p \in F^c \end{cases} \quad (3.11)$$

Luego, para una iteración en  $t > 0$ , los valores en el mapa de etiquetas  $(LABEL(p))_{p \in F}$  son actualizados usando la siguiente ecuación:

$$LABEL^{(t)}(p) = \min\{LABEL^{(t-1)}(p); \lambda; L_{min}(p)\} \quad (3.12)$$

Donde:

$$L_{min}(p) = \min\{LABEL^{(t-1)}(q)/q = (x_p + k, y_p + k) \in F\} \quad (3.13)$$

Y el parámetro  $\lambda > 0$  es un contador de componentes que se incrementa en 1 cada vez que el valor de  $LABEL(t)(p)$  se establece como  $\lambda$ . El valor del parámetro  $\lambda$  es siempre mayor en 1 que el número de componentes detectados. Este proceso es repetido hasta que no existan cambios que realizar en esta iteración. Al finalizar el proceso de etiquetado de los componentes, los pixeles  $p$  asociados con etiquetas  $LABEL(p)=0$  pertenece al fondo, es decir a  $F^c$ . Todos los pixeles  $p$  en un componente conectado dado de  $F$  están asociados con la misma etiqueta  $LABEL(p)>0$ .

## 4. Desarrollo de la Aplicación

### 4.1 Estructura de la Aplicación

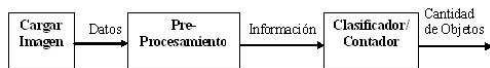


Figura 4.1: Diagrama en bloques de la Aplicación

El proceso que seguimos para obtener el número de células en las imágenes, se muestra en la figura 4.1, y la parte principal del mismo se

presenta en el procesamiento que efectuamos sobre las imágenes, ya que de esto depende la calidad del resultado del proceso.

Las imágenes que se presentarán en la siguiente sección, y con las cuales explicaremos el proceso, son las que obtuvimos empleando el detector de bordes Canny, sin embargo, nuestra aplicación adicionalmente, da la opción al usuario de utilizar también el detector de bordes Sobel.

### 4.2 Procesamiento de las imágenes: Explicación del Método 1

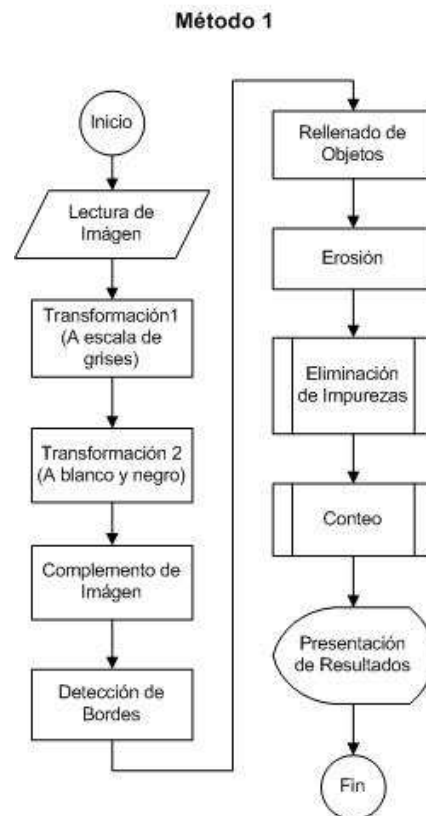
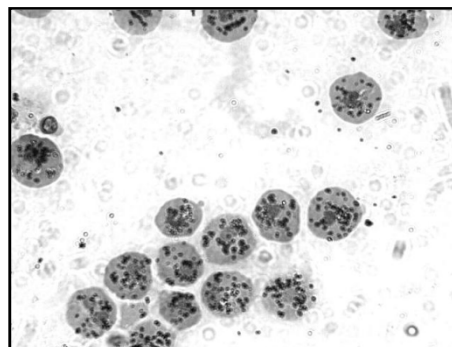


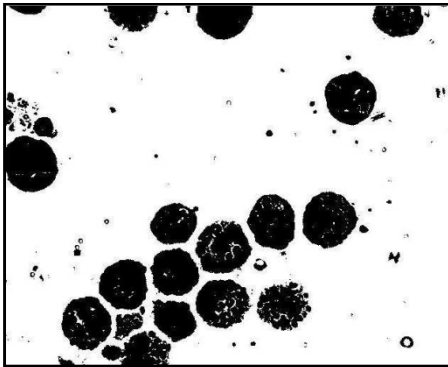
Figura 4.2: Algoritmo Método 1

A continuación se presentan los resultados obtenidos con cada paso del algoritmo 1:

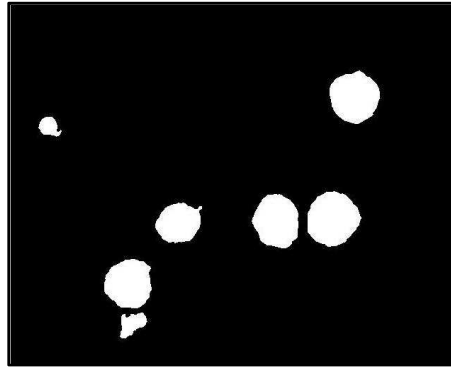
#### Imagen llevada a escala de grises



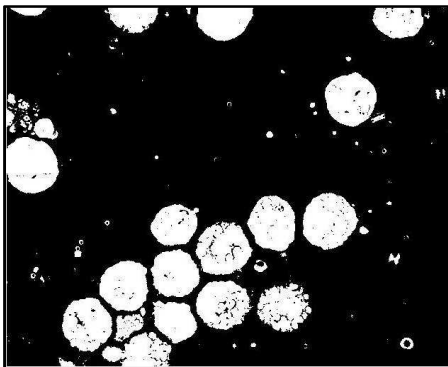
**Imagen en blanco y negro**



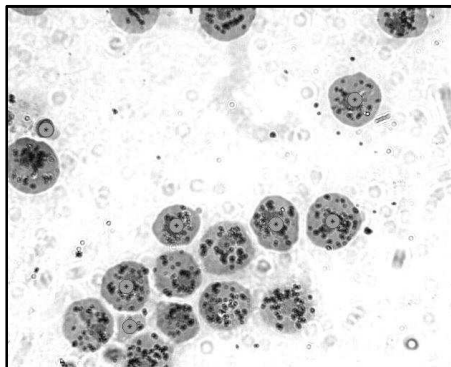
**Imagen segmentada mejorada**



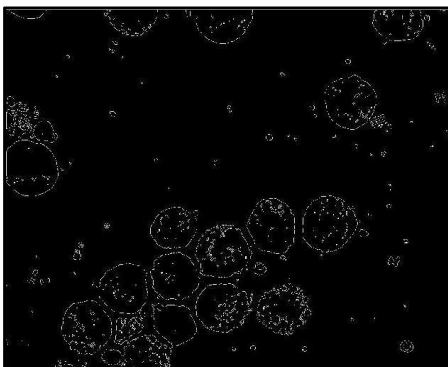
**Complemento de la imagen anterior**



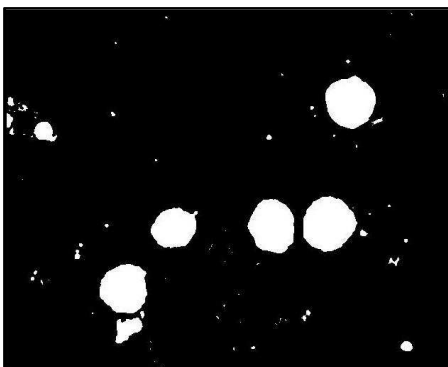
**Células contabilizadas con el Método 1**



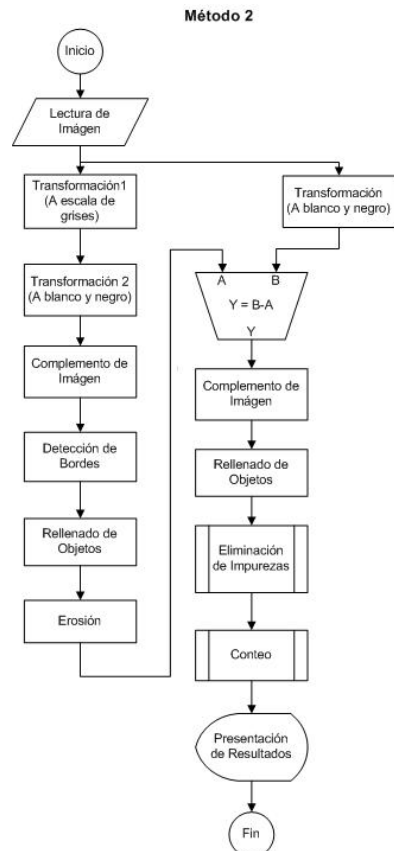
**Bordes detectados**



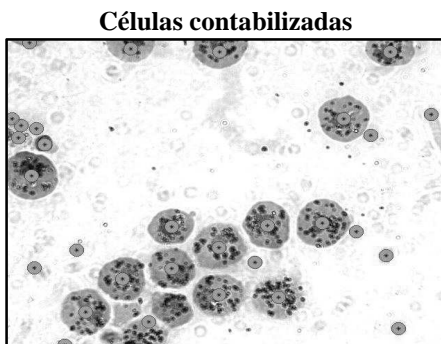
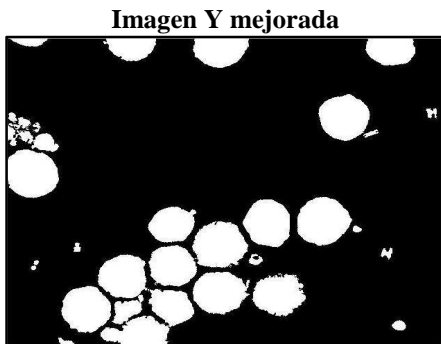
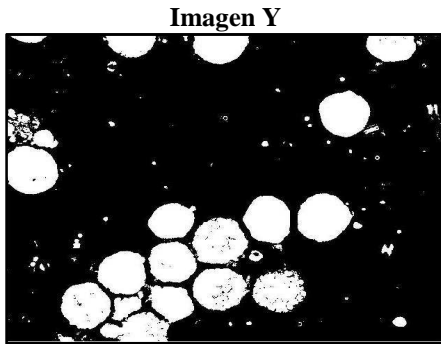
**Imagen segmentada**



## 4.2 Procesamiento de las imágenes: Explicación del Método 2



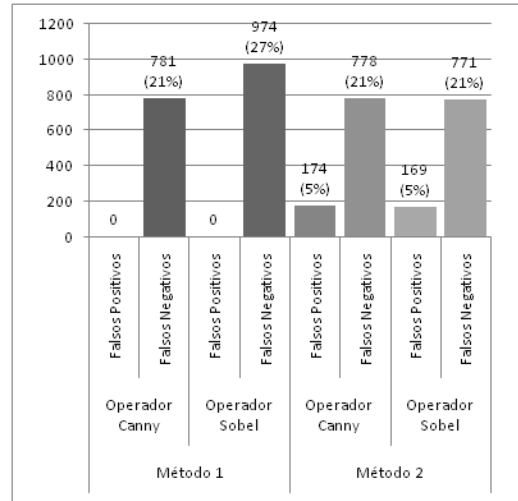
A continuación se presentan los resultados obtenidos con cada paso del algoritmo 2:



### 5. Resultados Obtenidos

Fueron analizadas 24 imágenes, de las cuales, en algunos casos la aplicación funcionó de mejor manera con un método que con el otro. En otros casos, ambos métodos arrojaban resultados muy parecidos, con una muy buena aproximación al número de células presentes en la imagen.

Imágenes Analizadas		24	
Número de células presentes en las imágenes		2031	
Método 1	Operador Canny	Falsos Positivos	0
		Falsos Negativos	781
Método 1	Operador Sobel	Falsos Positivos	0
		Falsos Negativos	974
Método 2	Operador Canny	Falsos Positivos	174
		Falsos Negativos	778
Método 2	Operador Sobel	Falsos Positivos	169
		Falsos Negativos	771



### 6. Conclusiones

1) De acuerdo a los resultados obtenidos, de entre las combinaciones posibles de métodos y operadores, la menor cantidad de errores se produjeron al utilizar el Método 1 junto con el operador Canny, ya que la suma de la proporción de falsos positivos y falsos negativos llega aproximadamente al 21%. Además Al emplear el Método 1 observamos que no existen falsos positivos (detección de objetos diferentes a los de nuestro interés) lo que le da a nuestra aplicación un cierto grado de selectividad.

2) En cuanto a la superioridad de los operadores, podemos señalar que el operador Canny mantiene una ventaja sobre el operador Sobel. Esta diferencia se nota sobre todo al momento de emplear el Método 1, en el cual la cantidad de falsos negativos (cuando el programa no detecta los objetos que nos interesan) es inferior al utilizar el operador Canny que al utilizar el operador Sobel.

3) Los falsos positivos que se registraron, fueron en algunos casos, elementos adicionales que se encuentran en las muestras juntos con las células, los cuales no pudieron ser removidos en su totalidad en la parte del procesamiento. Mientras que, la gran parte de falsos negativos, se debieron a que algunas células se encontraban agrupadas en las imágenes, lo que hizo que el programa detectara como una sola célula dichos grupos.

4) En lo que tiene que ver a la velocidad con que se obtiene el resultado del conteo utilizando nuestra aplicación, es evidente que esta, es mucho más veloz comparada con los procedimientos que existen en la actualidad. Sin embargo la exactitud del conteo realizado con nuestra aplicación todavía difiere con la que se obtiene en el conteo manual.

5) Finalmente, este trabajo no pretende reemplazar el procedimiento que se emplea actualmente en este tipo de análisis, solo deja planteada esta alternativa que puede servir como base de partida para futuros trabajos de investigación en la Medicina apoyados en el Procesamiento Digital de Señales.

## 7. Recomendaciones

1) La aplicación muestra un buen desempeño cuando se utilizan imágenes en las que las células se encuentran separadas o en imágenes con grupos de células, en cuyos grupos las células tienen bordes claramente diferenciables (caso de las células de cebolla), por lo que se recomienda que las imágenes a analizar tengan estas características.

2) La elección final del método de contabilización así como del operador depende del resultado que visualicemos al mandar a contar las células de la imagen. Es decir, que si observamos que al utilizar un método junto con un operador determinado, un mayor número de células es marcado, por obvias razones estos deberán ser el método y el operador elegidos.

3) El tamaño de las imágenes incide en la cantidad de recursos, sobre todo de memoria, empleados en el procesamiento por lo que es recomendable que estas no sean demasiado grandes, en nuestro caso obtuvimos buenos resultados con imágenes del orden de los 300Kb y la dimensión de las imágenes, promediaron los 600x400 píxeles.

4) Como trabajo futuro, se podría incluir el cálculo de otros parámetros de las células tales como el tamaño de las mismas. Adicionalmente, esta aplicación podría servir de base para contabilizar otros tipos de objetos tales como granos en general.

## Bibliografía

[1] BLANCHET Gerard, CHARBIT Maurice, "Digital Signal and Image Processing using Matlab", Iste, 2006.

[2] GONZALEZ Rafael C., WOODS Richards E., EDDINS Stevens L., "Digital Image Processing using Matlab", Prentice Hall, 2002.

[3] HUNT Brian, LIPSMAN Ronald, "A guide to Matlab for beginners and experienced users", Cambridge University Press, 2001.

[4] JAIN Anil K., "Fundamentals of Digital Image Processing", Prentice Hall, 1989.

[5] MARCHAND-MAILLET Stéphane, SHARAIA Yazid, "Binary Digital Image Processing", Academic Press, 2000.

[6] PRATT William K., "Digital Image Processing", Jhon Wiley & Sons, 2001

[7] SANCHEZ Omar, "Contar objetos en una imagen basado en grupos próximos", <http://omarsanchez.net/cuentaimg.aspx>, 2009