



Facultad de Ingeniería Eléctrica.

**Centro de Estudios de Electrónica y Tecnologías de la
Información (CEETI)**

TRABAJO DE DIPLOMA



**Título: Separación de agregados en imágenes de
microscopía celular.**

Autor: Liliana Guerrero Úbeda

Tutores: Dr. C. Juan Valentín Lorenzo Ginori

Ing. Lyanett Chinaa Valdés

Santa Clara

2011

"Año 53 de la revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

**Centro de Estudios de Electrónica y Tecnologías de la
Información (CEETI)**



TRABAJO DE DIPLOMA

**Título: Separación de agregados en imágenes de
microscopía celular.**

Autor: Liliana Guerrero Úbeda

Tutores: Dr. C. Juan Valentín Lorenzo Ginori

juanl@uclv.edu.cu

Ing. Lyanett China Valdés

lyanett@otn.vcl.cu

Santa Clara

2011

"Año 53 de la revolución"



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería Biomédica, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Jefe de Departamento

donde se defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

PENSAMIENTO

Siempre que te pregunten si puedes hacer un trabajo, contesta que sí y ponte enseguida a aprender cómo se hace.

Franklin Delano Roosevelt

DEDICATORIA

*A mi madre y mi tía por su inmenso amor y ser mis mayores fuentes de
energía e inspiración.*

A Dios por tantos milagros.

*A todas esas personas que confiaron en mí y me dieron fuerzas para
seguir adelante cuando pensaba que no podía.*

AGRADECIMIENTOS

Quiero agradecer sinceramente a muchas personas por haber contribuido en mi formación profesional y personal. Primero que nada, comenzaré agradeciendo:

A mi familia, por su apoyo incondicional a toda prueba, en especial a mi mamá y mi tía, por su inmensa paciencia y comprensión, por ser mis modelos de sabiduría, tenacidad y esfuerzo porque sin ellas no soy nadie y no hubiera llegado a ninguna parte.

A mi tutor, el Dr. Juan Lorenzo Ginori, por la oportunidad de trabajar bajo su asesoría en un área muy interesante y desconocida para mí; agradezco todas sus enseñanzas y sobre todo, su gran paciencia.

Le doy las gracias a la profesora María Esperanza por todos sus consejos, por animarme y demostrarme que siempre se puede dar un poquito más.

A mis amigas y amigos del pre y la universidad, Lina, Magdenia, Yalina, Idi, Saimy, Marisabel, Dayana, Meily, Anilet, Dayibi, Lyanett, Jelson, Heriberto, Sandor que me han hecho reír en tantas ocasiones y han llenado mi vida de historias.

A todos mis compañeros que he conocido durante la carrera que hicieron mi estancia en la universidad tan amena y memorable, y que compartieron conmigo estos inolvidables años.

A ellos y a todos aquellos que me acompañan, o me han acompañado, gracias.

TAREA TÉCNICA

1. Realizar una revisión bibliográfica sobre el tema segmentación de imágenes de microscopía celular utilizando técnicas de segmentación, operaciones morfológicas básicas y la extracción de características, con énfasis en las imágenes de eritrocitos de sangre humana.
2. Realizar un estudio detallado del problema de la detección y separación o subdivisión de los agregados en las imágenes de microscopía celular y de los algoritmos de PDI más representativos existentes para este fin.
3. Programar en Matlab y probar experimentalmente un algoritmo de detección de agregados basado en el procesamiento morfológico de imágenes en escala de grises.
4. Programar en Matlab y probar experimentalmente un algoritmo de detección de agregados basado en la clasificación de los objetos mediante el método *K-medias*.
5. Realizar un análisis estadístico comparativo del desempeño de ambos métodos partiendo de una primera segmentación realizada mediante los algoritmos de Otsu y de Lloyd.
6. Implementar un algoritmo de separación de los aglomerados basado en el empleo de la transformada *watershed*.



Firma del Autor



Firma del Tutor

RESUMEN

Desde hace años, el análisis automático de imágenes biomédicas está aumentando en importancia para la generación de diagnósticos y en la propia investigación médica. El uso de herramientas de procesamiento de imágenes es ahora esencial para analizar y comparar entre sí un gran número de imágenes procedentes de radiografías, microscopios, etc.

En los análisis de imágenes en microscopía celular (específicamente de los glóbulos rojos o eritrocitos en el Frotis de sangre) se hace necesario procesar grandes cantidades de imágenes para la realización de la clasificación y el conteo diferencial. Como este es un proceso que aún se desarrolla de forma manual, la automatización del mismo mediante la aplicación de técnicas de procesamiento digital de imágenes resulta una herramienta de gran aplicación. Para poder realizar un procesamiento de imágenes eficiente en este caso, se deben perfeccionar los métodos de detección y segmentación de agregados en las imágenes.

En este trabajo se implementan dos métodos de detección de agregados. El primero basado en procesamiento morfológico de imágenes, específicamente la transformada de distancia y la transformada *H-maxima* extendida. Mientras que el segundo método es utilizando el algoritmo *K-medias*. Luego de detectados los agregados para su separación se utilizará la transformada watershed con imposición de mínimos. Todo este proceso se realizará sobre una primera segmentación mediante el algoritmo de Otsu sobre la componente de intensidad de la imagen en el espacio HSV y mediante el algoritmo de Lloyd sobre la imagen a color.

TABLA DE CONTENIDOS

PENSAMIENTO.....	iv
DEDICATORIA	v
AGRADECIMIENTOS.....	vi
TAREA TÉCNICA	vii
RESUMEN	viii
INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTOS TEÓRICOS.....	5
1.1 Introducción a la Morfología y Segmentación en imágenes de microscopía celular ..	5
1.2 Operaciones morfológicas básicas.....	7
1.2.1 Erosión.....	8
1.2.2 Dilatación.....	8
1.2.3 Apertura y Cierre morfológicos	9
1.2.4 Transformada <i>H-maxima</i>	9
1.2.5 Imposición de mínimos	10
1.3 Concepto de Segmentación	10
1.4 Técnicas de Segmentación	11
1.4.1 Segmentación por comparación con un umbral.....	11
1.4.2 Algoritmo de Otsu.....	12

1.4.3	Algoritmo de Lloyd.....	13
1.4.4	Transformada <i>watershed</i>	13
1.4.5	Algoritmo <i>K-medias</i>	16
CAPÍTULO 2. MATERIALES Y MÉTODOS.....		18
2.1	Introducción	18
2.2	Herramientas utilizadas en el estudio	18
2.3	Características de las imágenes utilizadas	19
2.4	Descripción general del proceso de segmentación.....	20
2.5	Algoritmos de segmentación implementados en Matlab	26
2.5.1	Método de Otsu.....	26
2.5.2	Algoritmo de Lloyd.....	27
2.5.3	Detección de agregados utilizando la transformada de distancia y la transformada <i>H-maxima</i>	28
2.5.4	Algoritmo <i>K-medias</i> para separar en dos clases: agregados y no agregados .	30
2.5.5	Comparación entre los dos métodos propuestos de detección de agregados .	30
2.6	Segmentación de agregados empleando la transformada <i>watershed</i>	32
CAPÍTULO 3. RESULTADOS Y DISCUSIÓN		34
3.1	Resultados de la segmentación con el método de Otsu y el algoritmo de Lloyd	34
3.2	Resultados y comparación entre los experimentos propuestos para la detección de agregados	35
3.3	Validación estadística entre los experimentos propuestos para la detección de agregados	39
3.4	Segmentación fina de los eritrocitos aglomerados	40
CONCLUSIONES		42
RECOMENDACIONES		43

REFERENCIAS BIBLIOGRÁFICAS	44
ANEXOS	47
Anexo I Programas utilizados.....	47
Anexo II Tablas utilizadas	58

INTRODUCCIÓN

El veloz desarrollo y proliferación de las tecnologías de adquisición de imágenes médicas está revolucionando la medicina. Estas imágenes juegan un rol prominente en el diagnóstico y tratamiento de enfermedades, debido a que permiten a los científicos obtener información vital observando el interior del cuerpo humano de una forma no invasiva. La diversidad de principios de obtención de estas imágenes hace que existan varias modalidades de las mismas, entre las que aparecen las de resonancia magnética (MRI), tomografía axial computarizada (TAC o CT), ultrasonido (US), y microscopía celular, entre otras. Entre los algoritmos para la extracción de información a partir de imágenes tienen especial importancia los conocidos como algoritmos de segmentación, los que juegan un papel importante en numerosas aplicaciones biomédicas de tratamiento de imágenes.

Como principal motivación de este trabajo está precisamente la necesidad de analizar grandes cantidades de imágenes de microscopía celular, en las cuales usualmente aparecen aglomeraciones o agregados que es necesario identificar para su posterior segmentación. Este análisis es necesario en la clasificación y conteo de células (para el caso de los eritrocitos), que son de gran importancia en aplicaciones tales como la detección de patologías que se asocian con las alteraciones morfológicas de estos, y con la presencia de parásitos en su interior.

Los métodos para llevar a cabo la segmentación varían ampliamente dependiendo de la aplicación específica, tipo de imagen, y otros factores. El ruido, volúmenes parciales y movimientos en la imagen, también pueden tener consecuencias significativas en el desempeño de los algoritmos de segmentación. Actualmente no existe un método de segmentación que alcance resultados aceptables para todo tipo de imagen médica. No existen métodos que sean generales y que puedan ser aplicados a cualquier variedad de

datos. De cualquier forma, los métodos que son especializados para aplicaciones particulares pueden obtener mejores resultados tomando en cuenta un conocimiento a priori. Por lo tanto, la selección de un método apropiado para un problema de segmentación puede ser muy difícil.

En general puede decirse que un buen algoritmo de segmentación de imágenes será aquel que resalte de forma adecuada las regiones conectadas de interés en la imagen, de forma tal que estas regiones puedan ser utilizadas posteriormente en los procesos de análisis, reconocimiento de patrones y toma de decisiones.

El problema de la evaluación de la calidad en la segmentación es por tanto complejo, y ha de tener como punto de partida el disponer de métodos de segmentación que sean buenos candidatos para satisfacer los requerimientos planteados. En el caso del presente trabajo, se trata de implementar dos métodos de detección de agregados que resultan de un primer proceso de segmentación en el cual estos no se separan en sus componentes, y por último la segmentación destinada a subdividir dichos agregados. Los algoritmos se seleccionaron a partir de los resultados que se reportan en la literatura científica.

Por lo antes expuesto se propusieron los siguientes objetivos:

Objetivo General:

Desarrollar un programa de procesamiento digital de imágenes celulares en Matlab destinado a la detección y separación de agregados o aglomerados, aplicable al caso de imágenes de eritrocitos en sangre humana periférica.

Objetivos específicos:

1. Registrar, organizar y analizar información científico-técnica sobre los métodos de detección y separación de agregados en imágenes de microscopía celular empleando técnicas de Procesamiento Digital de Imágenes (PDI).
2. Estudiar y caracterizar algoritmos de PDI para la detección y separación de agregados en imágenes de microscopía celular.
3. Desarrollar programas en Matlab para la implementación de los algoritmos estudiados utilizando las operaciones morfológicas básicas y descriptores de rasgos.

4. Realizar pruebas experimentales con imágenes reales y evaluar la efectividad de los programas desarrollados.

Para lograr la solución del problema expuesto se planteó la siguiente tarea técnica:

1. Realizar una revisión bibliográfica sobre el tema segmentación de imágenes de microscopía celular utilizando técnicas de segmentación, operaciones morfológicas básicas y la extracción de características, con énfasis en las imágenes de eritrocitos de sangre humana.
2. Realizar un estudio detallado del problema de la detección y separación o subdivisión de los agregados en las imágenes de microscopía celular y de los algoritmos de PDI más representativos existentes para este fin.
3. Programar en Matlab y probar experimentalmente un algoritmo de detección de agregados basado en el procesamiento morfológico de imágenes en escala de grises.
4. Programar en Matlab y probar experimentalmente un algoritmo de detección de agregados basado en la clasificación de los objetos mediante el método *K-medias*.
5. Realizar un análisis estadístico comparativo del desempeño de ambos métodos partiendo de una primera segmentación realizada mediante los algoritmos de Otsu y de Lloyd.
6. Implementar un algoritmo de separación de los aglomerados basado en el empleo de la transformada *watershed*.

Los resultados de este trabajo son valiosos en el área de la imaginología médica relacionadas con el análisis de imágenes de microscopía celular. El campo en que se realiza esta investigación, está recogida en el artículo 131 de los lineamientos de la política económica y social del Partido y la Revolución, el cual hace alusión a sostener y desarrollar los resultados alcanzados en las ramas de la biotecnología, la producción médico-farmacéutica, la industria del software y el proceso de informatización de la sociedad, entre otros.

Este trabajo está estructurado de la siguiente forma: introducción, capitulario, conclusiones, recomendaciones, referencias bibliográficas y anexos. En el capítulo 1 se tratan los fundamentos teóricos de los métodos de separación de agregados y de segmentación propuestos y la morfología matemática básica. El segundo capítulo trata sobre la

implementación en Matlab de los algoritmos de detección de agregados y segmentación, así como la extracción de características definidas en el capítulo 1, aplicados a las imágenes de eritrocitos en muestras de sangre humana. El tercer capítulo se basa en la visualización y análisis de los resultados obtenidos con los métodos de detección y segmentación de agregados propuestos. Por último, en los anexos se pueden encontrar las líneas de comando de la programación realizada.

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS

1.1 Introducción a la Morfología y Segmentación en imágenes de microscopía celular

El procesamiento digital de imágenes es considerado como la transformación o el procesamiento de la imagen mediante una computadora digital, el cual abarca tareas tan diversas como la adquisición, digitalización, almacenamiento, visualización, transformación y comunicación.

En particular en el sector de la salud las imágenes se encuentran ampliamente difundidas, debido a la gran cantidad de equipos médicos emisores de imágenes que han sido desarrollados. La diversidad de principios de obtención de imágenes médicas hace que existan varias modalidades de las mismas, entre las que aparecen las de resonancia magnética (MRI), tomografía axial computarizada (TAC o CT), ultrasonido (US), y microscopía celular entre otros.

En el tratamiento de imágenes médicas es necesario desarrollar métodos automáticos de segmentación, que eviten en lo posible los procedimientos manuales. Los métodos manuales presentan dos dificultades importantes: el considerable esfuerzo de tiempo que precisan, y el error experimental intra-e inter-operador inherente. La repetitividad y rapidez de una segmentación automática es especialmente útil en las imágenes de microscopía celular ya que hace más eficiente y rápido el procesado de grandes volúmenes de imágenes, [1-3].

Durante los últimos 15 años, el campo de la microscopía óptica se ha desarrollado rápidamente con el advenimiento de una amplia gama de nuevas sondas (especialmente fluorescentes), y con la aparición de nuevos tipos de microscopios (microscopio confocal,

microscopía de fluorescencia, etc.), así como las aplicaciones de PDI (Procesamiento Digital de Imágenes). Todos estos avances han revolucionado nuestro conocimiento y entendimiento de las estructuras dinámicas y de los procesos relacionados con células vivas, [4], [5].

La segmentación de células es uno de los problemas más interesantes debido a la naturaleza compleja de los tejidos que las componen, así como a problemas inherentes a la técnica de microscopía que sea utilizada. Por ejemplo en la microscopía de vídeo, el desorden entre los objetos, el corto rango de los niveles de grises, y el bajo contraste, son graves dificultades que afectan a esta clase de imágenes, (ver Figura 1.1).

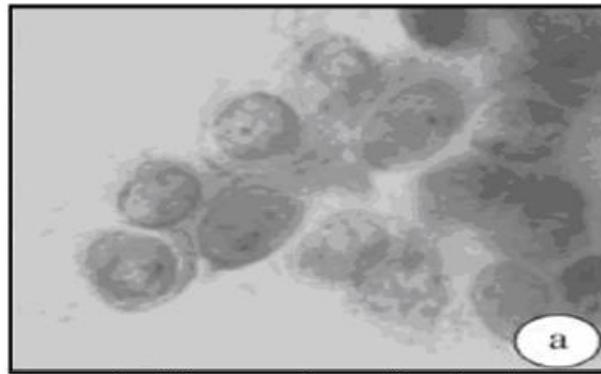


Figura 1.1: Imagen de células tomada mediante la microscopía de vídeo.

Con respecto a la microscopía de Inmunofluorescencia, el problema principal es dónde son perturbadas por la luz desenfocada, principalmente debido al espesor de la muestra. Esto puede conducir a algunas áreas donde se superponen varias células ya que no están en el mismo plano focal, (ver Figura 1.2).

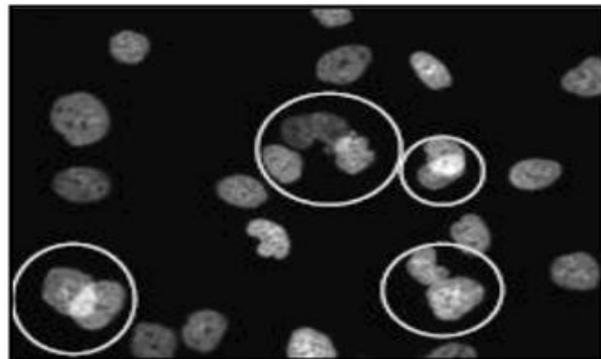


Figura 1.2 : Imagen de Inmunofluorescencia.

Este problema de superposición de áreas podría ser en gran parte solucionado usando el microscopio con un mismo foco, pero la adquisición de imágenes es mucho más lenta, por lo que es poco utilizado actualmente en pruebas de gran proyección en la medicina. Además, ambas fuentes de imágenes (confocal o microscopio fluorescente) presentan perturbaciones similares debido a la auto-fluorescencia del medio, el ruido de adquisición inducido por el detector de cámara, y varias imperfecciones del sistema óptico, [4], [5].

Se han desarrollado diversos estudios con el objetivo de segmentar células solapadas, ya que es una de las principales dificultades que enfrentan las imágenes de citología, permaneciendo como un tema abierto a la investigación científica y a la búsqueda de nuevas soluciones, debido a la importancia médica que de ella se deriva. Por lo tanto, el proceso de segmentación desarrollado para extraer cada célula y lograr una caracterización de forma confiable y una descripción de la estructura celular, es un reto que enfrenta hoy en día el procesamiento digital de imágenes en microscopía celular, cuyas soluciones definitivas no han sido todavía encontradas.

En este trabajo se hará énfasis en las operaciones morfológicas a realizar para la detección de agregados celulares (células que se tocan o superponen) con vistas a su posterior segmentación mediante la transformada *watershed*, [6], con la ayuda de Matlab. En este caso los agregados serían previamente segmentados (se denominará segmentación gruesa a esta etapa del proceso) empleando otras técnicas como pueden ser el método de Otsu para imágenes en escala de grises y el algoritmo de Lloyd para imágenes a color.

1.2 Operaciones morfológicas básicas

El procesamiento morfológico es una rama del Procesamiento Digital de Imágenes (PDI) que toma su nombre a partir de la morfología matemática, para extraer y procesar componentes de la imagen que resultan útiles en la representación y descripción de las formas (tales como límites, esqueletos y cascos convexos). La morfología matemática es una herramienta que sirve de base para la definición de diferentes filtros no lineales, de amplia utilización en el PDI. Las operaciones básicas en el filtrado morfológico son la erosión, la dilatación, la apertura y el cierre de una función, que en este caso es la imagen a filtrar, [6], [27]. El procesamiento morfológico de imágenes se aplica para las imágenes binarias y para las imágenes en escala de grises. En caso de estas últimas entre sus

aplicaciones se encuentran la transformada *H-maxima*, imposición de mínimos y la transformada de distancia las cuales serán de gran utilidad en este trabajo.

Para los procesos de dilatación y erosión – operaciones morfológicas básicas – se requiere entender primero las operaciones que se llevan a cabo durante estos procesos en el espacio . Estas operaciones son la reflexión y traslación de conjuntos. Las operaciones de reflexión y traslación de grupos se utilizan en muchas de las operaciones morfológicas. Estas forman operaciones basadas en un elemento estructurante (SE), pequeños conjuntos o subimágenes utilizadas para sondear una imagen bajo estudio en relación a propiedades de interés, [6], [27].

1.2.1 Erosión

En las imágenes binarias la erosión se define como se muestra en la ecuación (1).

$$(1)$$

Esta expresión indica que la erosión de A por B es el conjunto de todos los puntos z tales que B , trasladado por z , está contenido en A . En general, se asumirá que B es el elemento estructurante y A la imagen sobre la cual se desea aplicar el proceso de erosión. Mientras que en las imágenes en escala de grises la erosión se define como en la ecuación (2).

$$(2)$$

Donde f es la imagen a procesar y b el elemento estructurante es el dominio de la imagen y es el dominio del elemento estructurante. Se debe cumplir la condición de que $(s+x)$ y $(t+y)$ estén en el dominio f , x e y estén en el dominio de b , esta condición es análoga a la que se establece para el caso binario, en que elemento estructurante (SE) debe estar completamente contenido en el dominio de la imagen, [6], [27].

1.2.2 Dilatación

En las imágenes binarias la dilatación se define como se muestra en la ecuación (3).

$$(3)$$

Esta expresión indica que la dilatación de A por B es el conjunto de todos los desplazamientos z , tales que B y A se superpongan por lo menos en un elemento. Se asume una vez más que B será un elemento estructurante y A la imagen a someter a la operación.

Mientras que en las imágenes en escala de grises la dilatación se define como en la ecuación (4).

(4)

Donde f es la imagen a procesar y b el elemento estructurante, es el dominio de la imagen y es el dominio del elemento estructurante. Se debe cumplir la condición de que $(s-x)$ y $(t-y)$ estén en el dominio de f , x e y estén en el dominio de b , esta condición es análoga a la que se establece para el caso binario, en que los dos conjuntos (imagen y SE) deben solaparse en al menos un elemento, [6], [27].

1.2.3 Apertura y Cierre morfológicos

Las operaciones de apertura y cierre pertenecen, junto con las de erosión y dilatación, al conjunto de operaciones morfológicas básicas. Ambas comparten la similitud de estar compuestas por operaciones de erosión y dilatación, pero en diferente orden. Esto quiere decir que, la apertura de A por B es la erosión de A por B , seguida de la dilatación del resultado de nuevo por B , mientras que el cierre de A por B es la operación inversa, es decir, la dilatación de A por B , seguida de la erosión del resultado por B , utilizando en todos los casos un mismo elemento estructurante. En términos de sus efectos en las imágenes, la operación de apertura generalmente suaviza los contornos de los objetos, rompe istmos estrechos, y elimina protuberancias. La operación de cierre también tiende a suavizar las secciones de los contornos pero, por otra parte y en oposición a la apertura, generalmente fusiona pequeñas roturas y largos y estrechos golfos, elimina agujeros y rellena huecos en los contornos, [6], [27].

1.2.4 Transformada H -maxima

La transformada H -maxima se denota como es una operación morfológica que remueve cualquier domo (elevación) en la superficie de intensidad de la imagen que tenga altura menor o igual que h , y reduce la altura de los demás domos en una magnitud h . En la Figura 1.3 se muestra un ejemplo de esta operación. D es un elemento estructurante utilizado para implementar la función a partir de operaciones de reconstrucción morfológica. En Matlab, la función correspondiente es $Hm=imhmax(f, h)$, donde f es la función de intensidad de la imagen, [27].

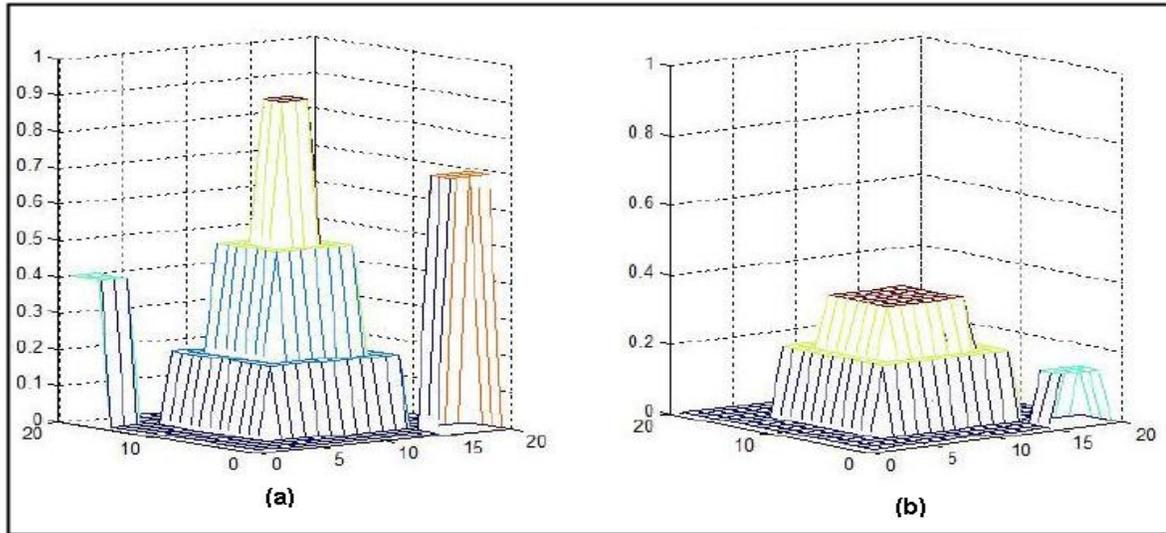


Figura 1. 3 : (a) Imagen original. (b) Transformada *H-maxima* con un umbral de 0.55.

1.2.5 Imposición de mínimos

Existen problemas en el procesamiento digital de imágenes donde la presencia de mínimos locales puede resultar un inconveniente. Ejemplo: el método de segmentación mediante la transformada *watershed*, que se tratará más adelante.

Cuando en una región existen mínimos locales en diferentes posiciones, es posible definir una posición en la que se imponga un mínimo regional (magnitud-) y que al mismo tiempo todos los demás mínimos locales sean suprimidos. Esta operación se realiza en Matlab mediante la función $I2=imimposemin(I, BW)$, que modifica la imagen de intensidad I mediante una reconstrucción morfológica, de modo que la imagen binaria BW (de iguales dimensiones que I) no sea igual a cero, [27].

1.3 Concepto de Segmentación

La segmentación es el proceso que subdivide una imagen en sus partes constituyentes u objetos, y es uno de los elementos fundamentales en el análisis automatizado de imágenes, debido a que es en esta etapa donde se extraen los objetos de interés para un procesamiento posterior, como puede ser su descripción y reconocimiento. El proceso de segmentación termina cuando los objetos de interés para una aplicación han sido aislados. En la práctica suelen distinguirse dos tipos de segmentación: de regiones o áreas y de objetos. La operación es común, lo que varía es aquello que se desea segmentar, [1-8].

1.4 Técnicas de Segmentación

En la literatura actual podemos encontrar una gran variedad de técnicas de segmentación, las cuales son de gran utilidad en imágenes de microscopía celular. Entre ellas podemos citar la transformada *watershed* y la segmentación por comparación con un umbral que posee un grado significativo de difusión, sobre todo en aplicaciones donde la velocidad es un factor importante.

En este trabajo se tratará la detección de agregados celulares, por una parte mediante la aplicación de la transformada de distancia y la transformada *H-maxima* y por otra mediante el algoritmo *K-medias*. A continuación se muestra una descripción resumida de algunas de las técnicas más utilizadas para esta primera etapa de segmentación, a la que se denominará en este trabajo segmentación gruesa, para diferenciarla de la separación de los agregados en sus componentes, que se denominará segmentación fina [1-8].

1.4.1 Segmentación por comparación con un umbral

Normalmente los métodos del valor umbral "binarizan" la imagen de partida, es decir se construyen dos conjuntos: el fondo de la imagen y los objetos buscados. La asignación de un píxel a uno de los dos conjuntos (0 y 1) se consigue comparando su nivel de gris g con un cierto valor umbral preestablecido t (en inglés *threshold*). La imagen final es muy sencilla de calcular ya que para cada píxel sólo hay que realizar una comparación numérica. La regla de cálculo correspondiente es:

$$(5)$$

Los métodos del valor umbral son métodos de segmentación completos, es decir cada píxel pertenece obligatoriamente a un conjunto y sólo a uno. Otros métodos de segmentación permiten que los conjuntos se solapen. Si en la imagen existen varios objetos con una luminosidad similar, con un mismo tono de gris, todos los píxeles que los componen pertenecerán al mismo conjunto. En la práctica siempre hay algún píxel que queda fuera del conjunto aunque pertenezca al objeto, normalmente debido a ruidos en la imagen original. En función del valor umbral que se escoja, el tamaño de los objetos irá oscilando [6], [23].

1.4.2 Algoritmo de Otsu

El algoritmo de segmentación de Otsu fue publicado en el año 1979 por Nobuyuki Otsu, [6]. Este método de detección de umbrales utiliza técnicas estadísticas, para analizar la dispersión de los niveles de grises. La importancia del método de Otsu radica en que es automático, ya que no necesita supervisión humana ni información previa de la imagen antes de su procesamiento.

El método de Otsu calcula el valor umbral de forma que la dispersión dentro de cada segmento sea lo más pequeña posible, pero al mismo tiempo la dispersión sea lo más alta posible entre segmentos diferentes. Para ello se calcula el cociente entre ambas varianzas y se busca un valor umbral para el que este cociente sea máximo, [6].

Como punto de partida tomamos dos conjuntos de puntos S_0 y S_1 , que serán definidos a partir del valor umbral T . T es la variable que buscamos, y los dos conjuntos son el resultado deseado en la segmentación. Sea P_g la probabilidad de ocurrencia del valor de gris g (G es el valor de gris máximo). Entonces la probabilidad de ocurrencia de los píxeles en los dos conjuntos es:

$$P_0 = \sum_{g=0}^T P_g \quad \text{y} \quad P_1 = \sum_{g=T+1}^G P_g \quad (6)$$

Si tomamos dos segmentos (o sea un solo valor umbral) la suma de estas dos probabilidades dará evidentemente 1. Si \bar{g} es la media aritmética de los valores de gris en toda la imagen, \bar{g}_0 y \bar{g}_1 los valores medios dentro de cada segmento, entonces se pueden calcular las varianzas dentro de cada segmento como:

$$V_0 = \sum_{g=0}^T (g - \bar{g}_0)^2 P_g \quad \text{y} \quad V_1 = \sum_{g=T+1}^G (g - \bar{g}_1)^2 P_g \quad (7)$$

La meta es mantener la varianza dentro de cada segmento lo más pequeña posible y conseguir que la varianza entre los dos segmentos sea lo más grande posible, [6]. El algoritmo de Otsu, aunque está concebido para su uso en imágenes en escala de grises, puede ser utilizado en las imágenes a color aplicándolo a sus diferentes componentes en un espacio de color dado.

1.4.3 Algoritmo de Lloyd

El algoritmo de Lloyd realiza una discretización o particionamiento del espacio, generando las llamadas regiones de Voronoi, que se definen mediante el uso de una medida de distancia, obteniéndose un conjunto de ejemplos representativos (prototipos o centroides). El objetivo del mismo es generar el conjunto de prototipos que minimice una determinada medida de distorsión o de distancia promedio, [9]. En la segmentación de imágenes, el algoritmo de Lloyd se aplica a los vectores de color que representan los píxeles en el espacio RGB, para formar agrupamientos o \mathcal{C} con estos según su color.

El algoritmo de Lloyd consta de los siguientes pasos:

1. Comenzar con un alfabeto inicial \mathcal{C}_0 , sea $\mathcal{C}_0 = \{c_1, \dots, c_k\}$.
2. Dado un alfabeto \mathcal{C}_i , ejecutar la iteración de Lloyd para generar un nuevo alfabeto \mathcal{C}_{i+1} .
3. Dado un alfabeto \mathcal{C}_i , subdividir el conjunto de entrada \mathcal{X} en particiones \mathcal{R}_i usando la siguiente condición:
$$x \in \mathcal{R}_i \iff \|x - c_i\| \leq \|x - c_j\| \quad \forall j \in \{1, \dots, k\} \quad (8)$$
 para todo $x \in \mathcal{X}$.
4. Calcular los centroides de cada partición para recalcular el alfabeto. Hacer $\mathcal{C}_{i+1} = \{c_1, \dots, c_k\}$. Si se generó una celda vacía en el paso 1, se asignará un vector alternativo (en vez del cálculo del centroide) para esa celda.

5. Calcular la distorsión media para \mathcal{C}_{i+1} . Si ha cambiado en una pequeña cantidad solamente, desde la iteración anterior parar, si no, hacer $i = i + 1$ e ir al paso 2, [9].

1.4.4 Transformada *watershed*

Análogamente a otras técnicas de segmentación, el objetivo de la transformada *watershed* es dividir en regiones la imagen de niveles de grises analizada. Generalmente una de ellas se corresponde con el fondo de la imagen y el resto con los objetos o regiones que se pretende extraer. El objetivo último de esta técnica es determinar los contornos que definen

dichos objetos. En este punto el problema es definir qué es contorno y que no lo es. Se considera que los contornos de una imagen se corresponden con las líneas donde el nivel de gris varía más rápidamente que en un determinado entorno vecino. Se define la imagen $grad(I)$, la cual está formada por píxeles cuyo valor es el módulo del gradiente en dicho punto (conocida como imagen gradiente). De esta manera, se considera que los contornos de la imagen original se corresponden con las líneas de cresta de la imagen gradiente, [6], [8].

- Algoritmo de la transformada *watershed*: El concepto de *watershed* se basa en visualizar una imagen en 3 dimensiones (3D): dos coordenadas espaciales vs. niveles de gris.

En esta interpretación 'topográfica', consideramos tres tipos de puntos:

1. puntos que corresponden a mínimos locales.
2. puntos en los que, si se coloca una gota de agua, esta cae con certeza en un único mínimo.
3. puntos en los que el agua caería con igual probabilidad en más de uno de estos mínimos.

Para un mínimo local particular, el conjunto de puntos que satisfacen la condición (2) se llama “*catchment basin* o *watershed* o cuenca” de este mínimo. Los que satisfacen la condición (3) forman líneas de cresta en la superficie topográfica y son llamadas “líneas divisorias o líneas de *watershed* o líneas de cresta”, (ver Figura 1.4). El objetivo principal de los algoritmos de segmentación es detectar estas líneas divisorias o de cresta, [8].

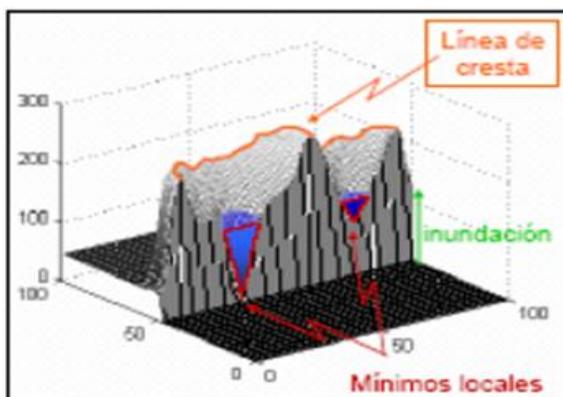


Figura 1. 4: Línea de cresta.

La idea básica es la siguiente: supongamos que se hace un pequeño agujero en cada mínimo local, y que todo el relieve topográfico es inundado desde abajo, dejando que el agua entre a velocidad constante. El agua va subiendo e inundando las cuencas. Cuando el agua de dos cuencas está a punto de juntarse, se construye un dique para evitar la fusión. La inundación continúa, y llega a un punto en que solo se ve la parte superior de los diques por encima de la línea de agua. Las líneas de *watershed* forman un camino conexo, dando por lo tanto bordes continuos entre las regiones. Por otro lado, el uso de la imagen gradiente, para aplicar directamente sobre ella la técnica de *watershed* generalmente produce el efecto conocido con el nombre de sobresegmentación. Este efecto provoca que los verdaderos contornos queden enmascarados por una infinidad de contornos falsos o irrelevantes. Esto es así aunque se haya tomado la precaución de filtrar previamente la imagen original o su correspondiente imagen gradiente [6], [8].

En el caso particular de aplicación de la transformada *watershed* a la separación de agregados, que es el caso que nos ocupa, se utiliza comúnmente en lugar del gradiente la transformada de distancia, la cual se define como: la distancia desde cada píxel hacia el píxel más cercano diferente de cero. La transformada de distancia puede ser calculada usando la función de Matlab *bwdist*, cuya sintaxis es:

$$D=bwdist(f)$$

En la Figura 1.5 (a) se muestra una matriz de una pequeña imagen binaria y en la Figura 1.5 (b) se muestra su transformada de distancia correspondiente, [6].

1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
0	0	0	0	0	1.00	1.00	1.41	2.00	2.24
0	0	0	0	0	1.41	1.00	1.00	1.00	1.41
0	1	1	1	0	1.00	0.00	0.00	0.00	1.00

(a)

(b)

Figura 1. 5: (a) Pequeña imagen binaria. (b) Transformada de distancia.

1.4.5 Algoritmo *K-medias*

El algoritmo *K-medias* es una de las heurísticas comúnmente utilizadas para resolver el problema de obtener aglomeraciones o *clustering*. El algoritmo de Lloyd antes descrito para la segmentación de imágenes a color no es más que el algoritmo *K-medias* en el que se sigue una estrategia de particiones para obtener los centroides y regiones por etapas, [6]. A continuación se describe un algoritmo *K-medias* para el caso particular de trabajar con un espacio de características de dimensión igual a 1. (Por ejemplo los niveles de gris de una imagen monocromática.)

Suponiendo que tenemos k clases, debemos elegir $k-1$ umbrales y k medias de las clases:

1. En primer lugar, se eligen los K valores iniciales, para las medias de cada una de las clases, o centroides (que pueden ser, por ejemplo una muestra de cada clase, esto es, un nivel de gris que suponemos corresponde a cada clase).
2. Calculamos los umbrales utilizando un criterio de distancia mínima con respecto a los centroides.
3. Calculamos la media de cada una de las clases, a partir de los umbrales anteriores.
4. Repetimos los pasos 2 y 3, hasta que se cumpla algún criterio de terminación, como por ejemplo que la diferencia entre los valores promedio consecutivos de las muestras en cada clase (centroides) sea menor que cierto valor para todas las clases o grupos.

Los principales inconvenientes del algoritmo *K-medias* son:

1. El agrupamiento final depende de los centroides iniciales.
2. La convergencia en el óptimo global no está garantizada, y para problemas con muchos ejemplares, requiere de un gran número de iteraciones para converger.

Un factor que afecta en gran medida el costo computacional del algoritmo *K-medias* es el número de iteraciones que necesita realizar, ya que por cada iteración calcula la distancia de cada objeto a los centroides de los grupos, [6].

En este trabajo el algoritmo *K-medias* se utiliza para separar en dos clases (agregados y no agregados) utilizando diferentes rasgos de la función *regionprops* del Matlab, sobre esta función y sus diferentes características se tratará mas adelante.

CAPÍTULO 2. MATERIALES Y MÉTODOS

2.1 Introducción

La identificación automática de las aglomeraciones o agregados (células que se superponen o se tocan) para proceder a su segmentación constituye un importante problema en la imaginología celular, (ver Figura 2.1 (a) y (b)). En años recientes, muchas técnicas han sido propuestas para la segmentación y procesamiento morfológico de las células, las que frecuentemente dependen del tamaño, la forma y el color de las células a segmentar,[15-20], [23-26].

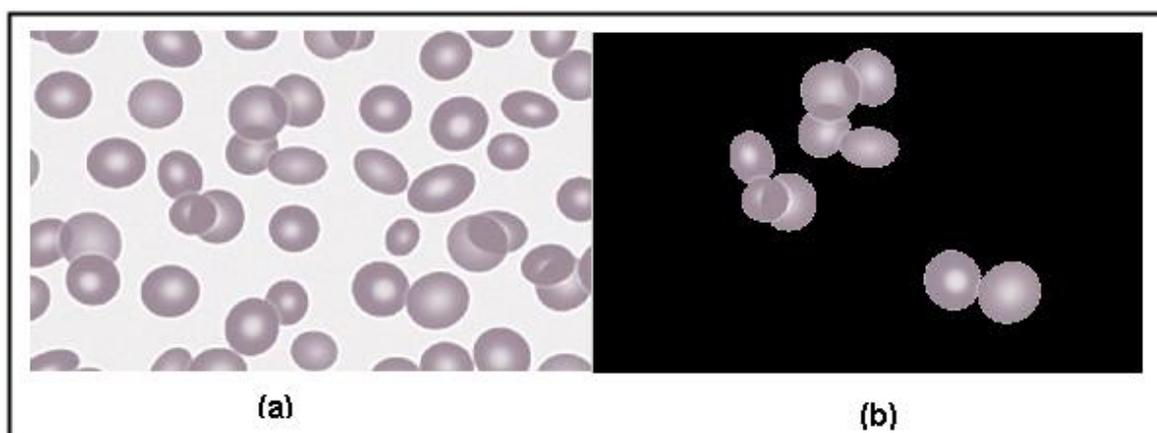


Figura 2. 1: (a) Imagen de células (eritrocitos). (b) Células aglomeradas.

2.2 Herramientas utilizadas en el estudio

La plataforma principal utilizada durante esta investigación ha sido el software Matlab y su correspondiente Toolbox de procesamiento de imágenes. En Matlab una imagen en escala de grises es una matriz de píxeles $M \times N$ donde cada elemento representa una intensidad de luz. Las funciones principales de Matlab utilizadas en este trabajo han

lado: *rgb2hsv*, *graythresh*, *im2bw*, *imclearborder*, *imfill*, *bwareaopen*, *bwlabel*, *regionprops*, *kmeans*, *cat*, *bwdist*, *imextendmax*. Estas funciones se explicarán con más detalle en los epígrafes siguientes. También se utilizó el programa estadístico SPSS 11.5 para comprobar cuán eficaces fueron los dos métodos de detección de agregados implementados con respecto a las medias aritméticas, las medianas y la varianza de la sensibilidad (Se) y especificidad (Sp) calculadas a cada una de las imágenes segmentadas por el algoritmo de Otsu o Lloyd. Los resultados obtenidos se encuentran explicados en el capítulo 3.

2.3 Características de las imágenes utilizadas

Las imágenes utilizadas para los estudios realizados en este trabajo fueron tomadas de un atlas de hematología que aparece en Internet el cual consta de 24 imágenes y 4 imágenes sintéticas simuladas con un programa desarrollado en el CEETI. Las imágenes son obtenidas sobre un extendido de sangre humana y se encuentran todas en el espacio de color RGB. En la Tabla 2.1 se muestran los datos de 20 de las imágenes del atlas y las 4 sintéticas, con las cuales se trabajó, con el objetivo de ilustrar la diversidad presente en las mismas, [11], [12].

Tabla 2.1: Datos de 13 de las imágenes utilizadas.

Nombre de la imagen	Tipo de imagen	Tamaño en píxeles	Formato de la imagen	Tinción utilizada	Aumento
Imagen	Sintética	500x750	BMP	-	-
Imagen2	Sintética	500x750	BMP	-	-
Imagen65	Sintética	500x750	BMP	-	-
Eritrocitos sintéticos	Sintética	338x510	JPG	-	-
Elliptocyte	Real	508x748	JPG	May-Giemsa	x1000
Imagen4	Real	750x500	BMP	May-Giemsa	x1000
Img04	Real	498x744	JPG	May-Giemsa	x1000
Spherocyte	Real	508x747	JPG	May-Giemsa	x1000
Tear Drop Cell	Real	508x748	JPG	May-Giemsa	x1000
Rouleaux formation	Real	747x508	JPG	May-Giemsa	x1000
Pappenheimers body	Real	506x747	JPG	May-Giemsa	x1000
Imag1	Real	569x376	JPG	May-Giemsa	x1000
Imag2	Real	570x379	JPG	May-Giemsa	x1000
Imag3	Real	570x380	JPG	May-Giemsa	x1000
Imag4	Real	569x378	JPG	May-Giemsa	x1000
Imag5	Real	569x378	JPG	May-Giemsa	x1000
Imag6	Real	569x378	JPG	May-Giemsa	x1000
Imag7	Real	567x377	JPG	May-Giemsa	x1000
Imag8	Real	570x379	JPG	May-Giemsa	x1000
Imag9	Real	569x376	JPG	May-Giemsa	x1000
Imag10	Real	571x379	JPG	May-Giemsa	x1000
Imag11	Real	568x378	JPG	May-Giemsa	x1000
Imag12	Real	570x376	JPG	May-Giemsa	x1000
Imag13	Real	569x379	JPG	May-Giemsa	x1000

2.4 Descripción general del proceso de segmentación

Para el desarrollo de la segmentación se siguió un esquema general, el cual se muestra en la Figura 2.2.

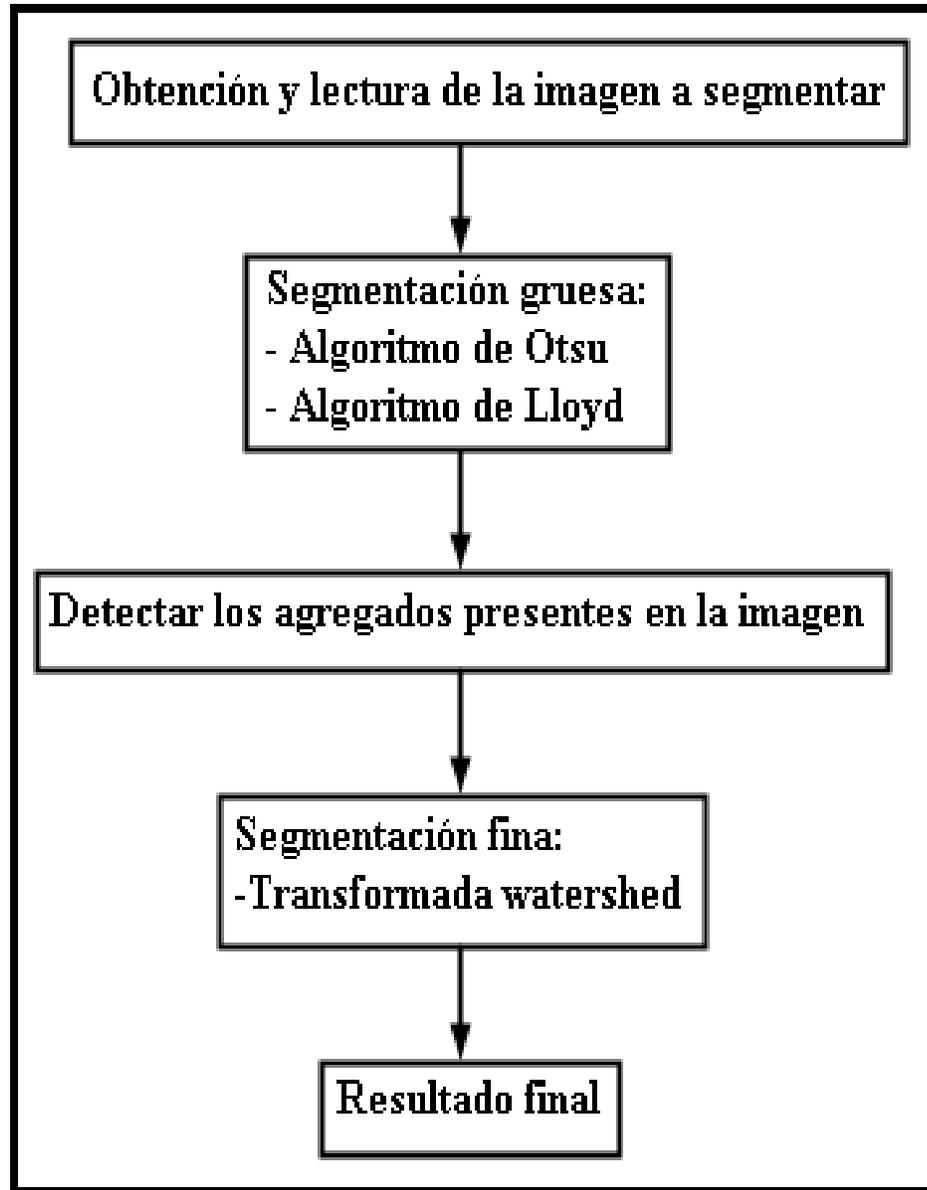


Figura 2. 2: Diagrama del algoritmo utilizado en el proceso de segmentación.

Primeramente se debe cargar al Matlab la imagen a segmentar con la función *imread*, como las imágenes están en el espacio de color RGB, es necesario llevarlas al espacio de color HSV, lo cual se realizó con la función *rgb2hsv* de Matlab, (ver Figura 2.3 y Figura 2.4).

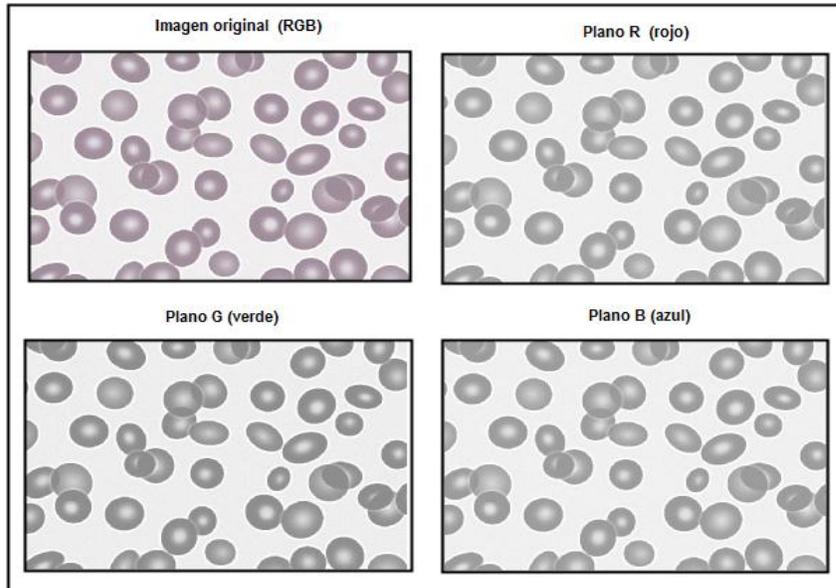


Figura 2.3: Imagen a color con sus componentes en el espacio de color RGB.

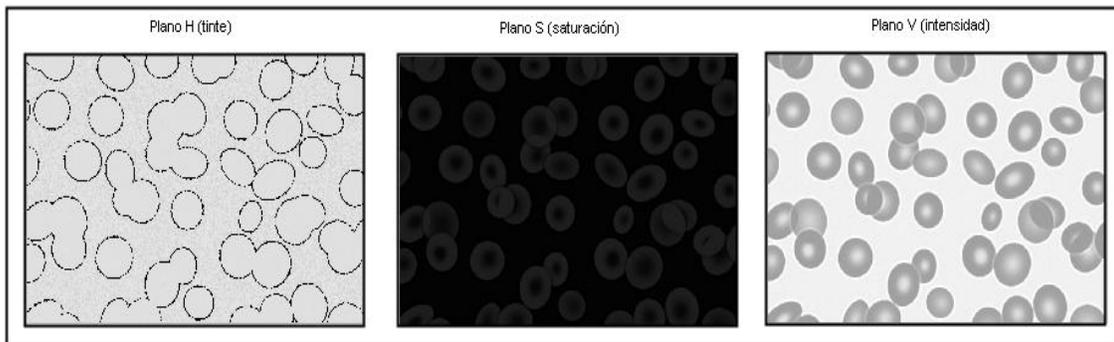


Figura 2.4: Imagen a color con sus componentes en el espacio de color HSV.

Luego se realiza lo que se llamó una segmentación gruesa, consistente en un primer proceso de segmentación en el cual no es posible separar los agregados en los componentes que lo componen. Para realizar la segmentación gruesa se implementaron en este trabajo el método de Otsu utilizando las imágenes de intensidad y el algoritmo de Lloyd para las imágenes a color en el espacio RGB.

La evaluación en forma cuantitativa del desempeño de estos métodos de segmentación no formó parte de los objetivos de este trabajo, ya que los resultados de los procesos de segmentación implementados solo fueron utilizados a los efectos de detectar la presencia o no de agregados celulares, y dado que ambos algoritmos han sido descritos en la literatura como efectivos en diferentes situaciones, [6], [28].

En el caso del método de Otsu se considera más favorable la imagen V (es el máximo de las tres componentes HSV) como información relativa a la intensidad para la aplicación de un algoritmo basado en umbrales, dado que H es un ángulo por lo que sus valores se distribuyen en forma circular, S es la saturación con la cual se trabajó anteriormente y no se obtuvieron buenos resultados. No obstante, el programa implementado permite aplicar el algoritmo de Otsu a cualquier imagen en escala de grises derivada de la imagen a color, como pueden ser las componentes R, G y B (espacio RGB), o H, S y V (espacio HSV).

Con la función *graythresh* se determina el umbral global de la imagen de saturación y con la función *im2bw* se convierte la imagen a binaria, sobre la base del umbral calculado, todas estas son funciones de Matlab, ver Figura 2.5 (a), (b) y (c).

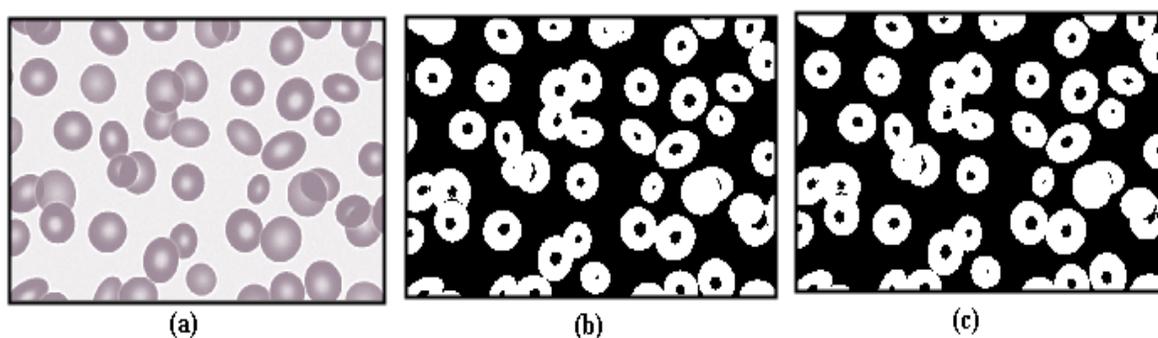


Figura 2.5: (a) Imagen original ‘Imagen 2’. (b) Imagen segmentada por el método de Otsu. (c) Imagen segmentada por el método de Lloyd.

Debido a la forma que presentan los eritrocitos, de ser células que no tienen núcleo cuando maduran y con una depresión en el centro, al aplicarse los algoritmos de segmentación se evidencia la presencia de “huecos” en el centro de las células, los cuales se rellenan mediante la función *imfill* de Matlab. En esta función un “hueco” se analiza como un conjunto de píxeles del fondo que no pueden ser alcanzados si se rellena el fondo a partir de los bordes de la imagen. Después se procede a eliminar las células que tocan los bordes con la función de Matlab *imclearborder*, basado en que estas células no aportan una información morfológica apropiada, ya que las mismas aparecen recortadas en alguna medida, como se muestra en la Figura 2.6.

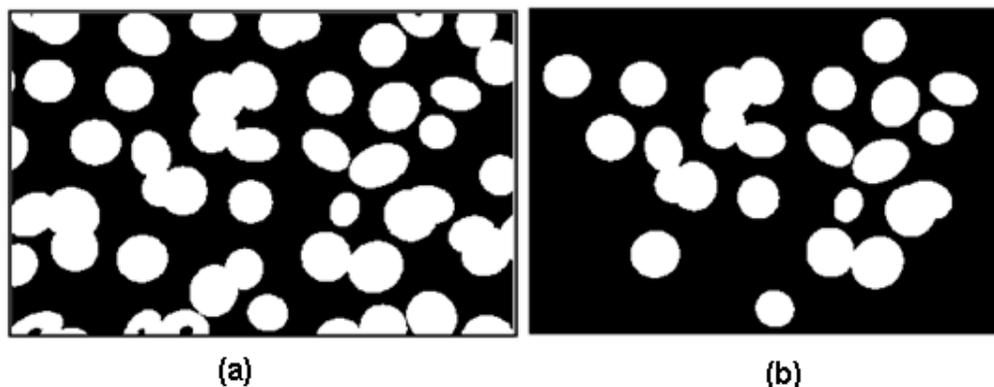


Figura 2. 6: (a) Imagen con los huecos rellenos. (b) Imagen con los bordes eliminados.

En las imágenes pueden aparecer, luego de realizar la segmentación, de rellenar los “huecos” y eliminar las células que tocan los bordes, elementos de pequeño tamaño que no forman parte de la estructura celular y que constituyen artefactos o ruido. Para eliminarlos se aplicó la función *bwareaopen* que permite eliminar objetos cuya área sea inferior a cierto umbral y mantener inalterados los demás objetos. Este umbral se determinó de forma experimental a partir del análisis del conjunto de imágenes.

Una vez realizados los procesos anteriormente descritos, es posible pasar a realizar la detección de agregados, para ello primeramente se descompone la máscara de la imagen correspondiente, en regiones conectadas etiquetadas con la función *bwlabel*, propia de la librería de procesamiento de imagen de Matlab, cuya sintaxis es:

$$[L, num]=bwlabel (BW, n)$$

donde *BW* es la imagen con la región de interés binarizada y *n* es la vecindad, pudiendo ser de vecindad a 8 o a 4. El resultado es una imagen *L* con cada componente conectada marcada con una etiqueta y *num* es el número de componentes conectadas en la imagen. El etiquetado permite identificar los objetos que serán clasificados como agregados o células simples, de acuerdo con los dos métodos implementados en este trabajo, ver Figura 2.7.

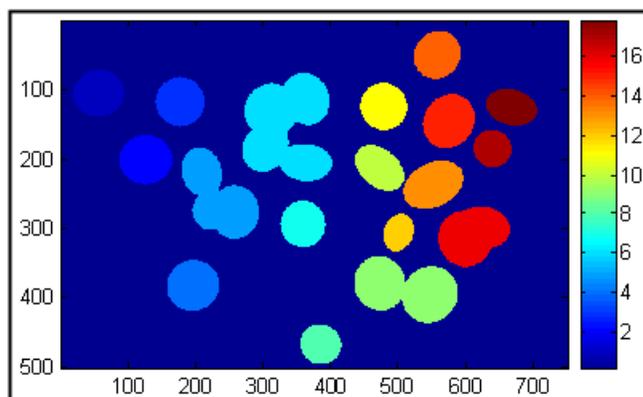


Figura 2.7: Imagen etiquetada.

El color azul oscuro representa las regiones de menor valor de la etiqueta y el color rojo oscuro representa las regiones de mayor valor.

La función principal que se encarga de realizar el estudio de las características de cada región de la imagen es una función de la biblioteca de procesamiento de imagen de Matlab, esta función se conoce como *regionprops*, la cual necesita como parámetro de entrada la matriz etiquetada (L) que es la imagen resultado y las propiedades que queremos que calcule (*properties*). La sintaxis es la siguiente,

$$STATS = \text{regionprops}(L, \text{properties})$$

La función *regionprops* devuelve una estructura *STATS* con todas las medidas que hemos introducido como parámetro de entrada en *properties*.

Las propiedades de la función *regionprops* utilizadas fueron:

- *Area*: Escalar; es el número actual de píxeles en la región.
- *ConvexArea*: Escalar; esta medida se corresponde con el área de la envoltura convexa, que es el número de píxeles que ésta contiene.
- *Centroid*: Escalar; el primer elemento del centroide es la coordenada horizontal (o x-coordenada) del centro de masa, y el segundo elemento es la coordenada vertical (o y-coordenada).
- *EquivDiameter*: Escalar; es el diámetro de un círculo con la misma área que la región.
- *Extent*: Escalar; es la proporción de los píxeles en el rectángulo base que también están en la región. Se calcula como el área dividida por el área del rectángulo base.

- *FilledArea*: Escalar; es el número de píxeles en FilledImage (imagen binaria (lógica) del mismo tamaño del rectángulo base de la región. Los píxeles ON corresponden a la región, con todos los huecos rellenos).
- *Solidity*: Escalar; es la proporción de los píxeles en la casco convexo que también está en la región. Se calcula como el $\text{Área}/\text{ConvexArea}$.

Con todos estos rasgos podemos construir una matriz (MxN) que contenga por las filas la cantidad de objetos etiquetados y por las columnas los rasgos anteriormente descritos.

Una vez determinados los agregados presentes en la imagen estos necesitan otro procesamiento para su separación siendo este la segmentación fina (obliga a tomar un decisión en cuanto a si el píxel está dentro o fuera del objeto). La misma consiste en segmentar las aglomeraciones en las células (eritrocitos) constituyentes, aplicando la segmentación mediante la transformada *watershed*. Se obtiene el resultado final que es la separación de los agregados para su utilización en procesos posteriores. Su programación se puede observar en el anexo (I).

2.5 Algoritmos de segmentación implementados en Matlab

A continuación se explican los pasos de los algoritmos que han sido implementados en Matlab para el proceso de segmentación gruesa (método de Otsu y algoritmo de Lloyd) y para la detección de agregados (transformada de distancia junto a la transformada *H-maxima* y el algoritmo *K-medias*) en las imágenes.

2.5.1 Método de Otsu

Para la implementación del método de Otsu, utilizando la información relativa a la intensidad (plano V), han sido utilizadas las imágenes de intensidad para la aplicación de un algoritmo basado en umbrales. Con la función *graythresh* se determina el umbral global de la imagen de intensidad y con la función *im2bw* se convierte la imagen a binaria, sobre la base del umbral calculado, todas estas son funciones de Matlab y su programación se puede observar en el anexo (I).

2.5.2 Algoritmo de Lloyd

El diagrama del algoritmo de Lloyd implementado en Matlab se muestra en la Figura 2.8, la función que se programó en Matlab con dicho algoritmo se nombra *distRGB* y su programación se puede ver en el anexo (I). Para la implementación del algoritmo se siguieron los pasos del epígrafe 1.4.3.

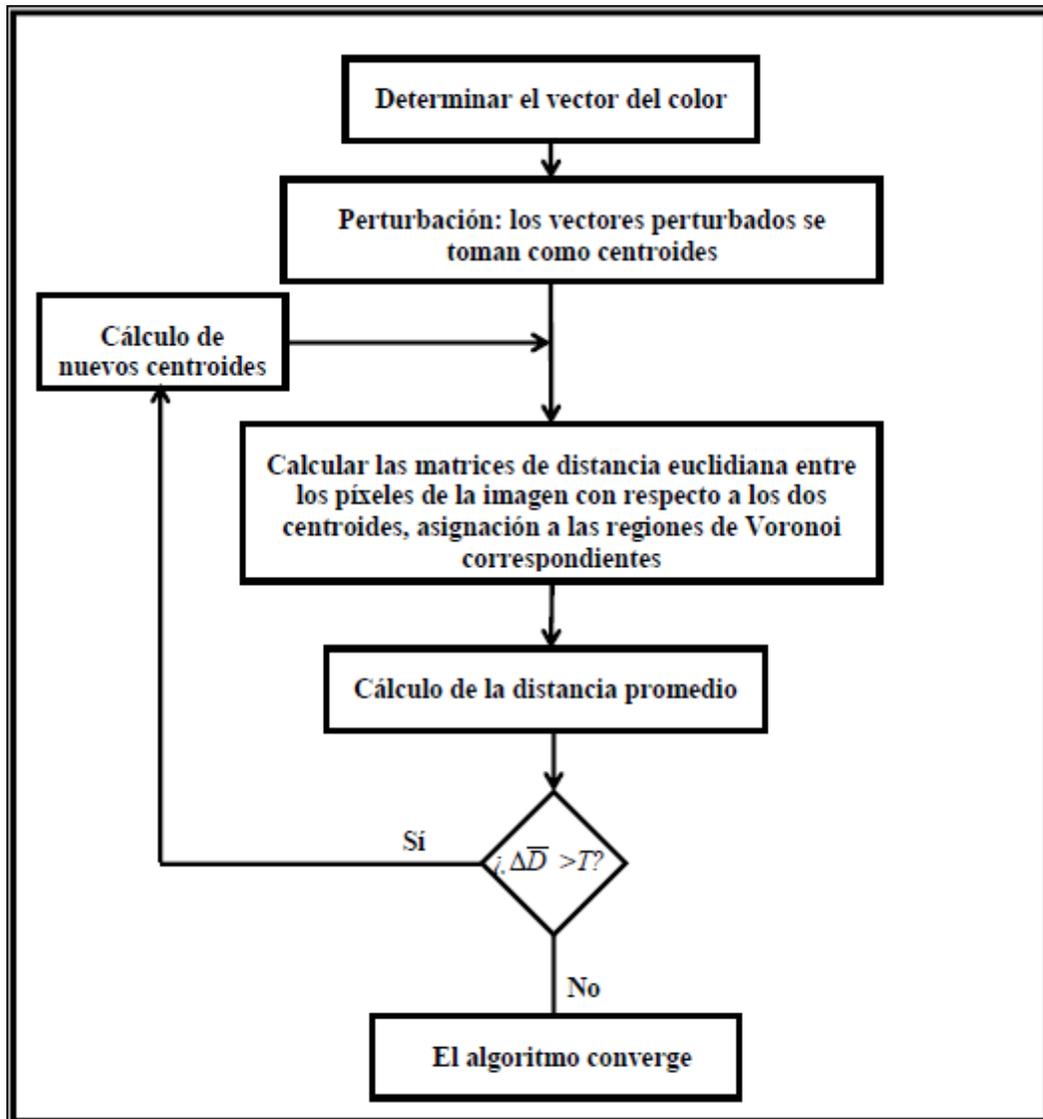


Figura 2.8: Diagrama del algoritmo de Lloyd.

2.5.3 Detección de agregados utilizando la transformada de distancia y la transformada *H-maxima*

El diagrama de flujo del algoritmo implementado en Matlab se muestra en la Figura 2.9. El programa con dicho algoritmo se puede ver en el anexo (I). Para la implementación del algoritmo se siguieron los siguientes pasos:

1. Etiquetar la imagen segmentada por el método de Otsu o el de Lloyd.
2. Obtener la transformada de distancia del primer objeto.
3. Obtener un primer valor de los máximos extendidos de la transformada *H-maxima*. Proceso que se repetirá sucesivamente.
4. Si $DNUM > 1$ (cantidad de máximos extendidos en la transformada *H-maxima*) comienza un ciclo *while* con el criterio de parada $cont \leq 5$ (este valor se determinó experimentalmente y se puede cambiar). El contador *cont* se inicializa en 0 al inicio del programa, es el encargado de contar la cantidad de veces que se repite el número de máximos. En este ciclo se aumenta el valor de *j* (es el umbral) el cual se inicializa en cero al inicio del programa para mostrar todos los puntos máximos que en realidad existen en el objeto. El aumento del umbral es $\Delta j = 0.001$ (se puede variar este valor) y se vuelven a calcular las transformadas de distancia y *H-maxima* extendida.
5. Si $DNUM = A(h)$ se incrementa *cont* sino $cont = 0$. El contador *h* se inicializa al inicio del programa en 1 (es la primera posición) ya que es el encargado de la posición de los valores guardados en el vector *A*.
6. Se aumenta *h* en 1 y se ubica el nuevo valor de *DNUM* en el vector *A* para su posterior comparación con el valor anterior de *DNUM*.
7. Si el último valor guardado en el vector $A > 1$ entonces hay un agregado el cual se guarda en una matriz de ceros llamada *aglom*. Los puntos máximos de los agregados detectados también se guardan en una matriz de ceros llamada *ptomax*.
8. Cuando los agregados son detectados o es un solo objeto se vuelve al inicio del programa para hacerle las mismas operaciones al siguiente objeto etiquetado.

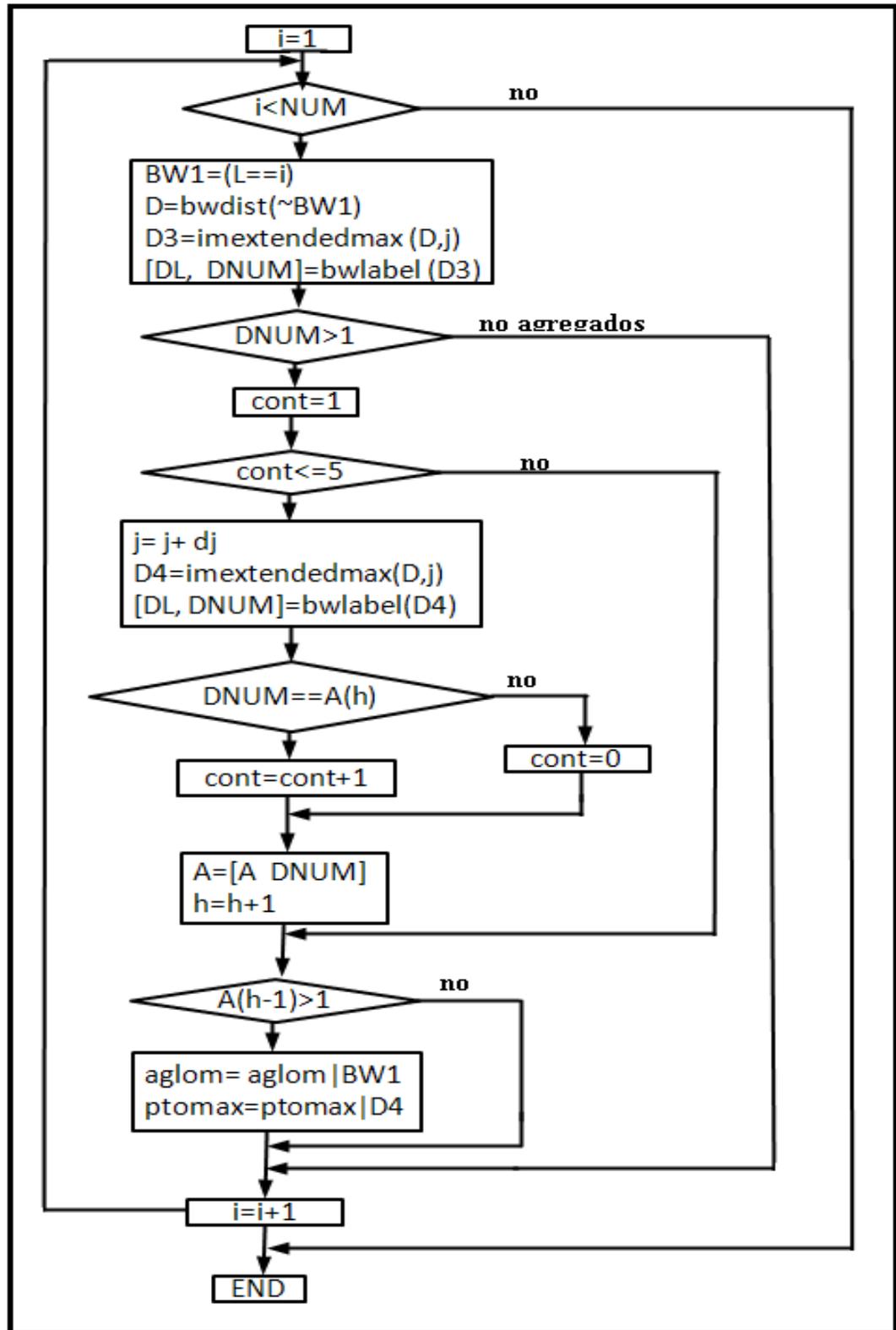


Figura 2. 9: Diagrama de flujo para detectar agregados utilizando la transformada de distancia y la transformada *H-maxima*.

2.5.4 Algoritmo *K-medias* para separar en dos clases: agregados y no agregados

El diagrama del algoritmo *K-medias* implementado en Matlab se muestra en la Figura 2.10 que se encuentra en la página 31. El programa con dicho algoritmo se puede ver en el anexo (I). Para la implementación del algoritmo se siguieron los siguientes pasos:

1. Etiquetar la imagen segmentada por el método de Otsu o el de Lloyd.
2. Calcular con la función *regionprops* los rasgos descritos anteriormente en el capítulo 1. Estos rasgos fueron escogidos porque dieron los mejores resultados con respecto a los demás.
3. Realizar una matriz para concatenar todos los rasgos con la función *cat*.
4. Aplicar la función *kmeans* a dicha matriz y dividir en dos clases (agregados y no agregados).

2.5.5 Comparación entre los dos métodos propuestos de detección de agregados

La comparación entre los algoritmos de detección de agregados en cuanto a su desempeño en el proceso de detección se desarrolla en el capítulo 3 sobre la base de un análisis estadístico. Este análisis se realizó con la prueba de rango con signo de *Wilcoxon*, [29], donde se formuló la hipótesis nula de que (medianas) contra la hipótesis alternativa que . Si el valor observado de (región crítica) es menor que el valor crítico tabulado, se rechaza para un nivel de significancia () de 0.05.

Con esta prueba detectamos si verdaderamente el método de detección de agregados utilizando la transformada de distancia y transformada *H-maxima* es mejor que el algoritmo *K-medias* para separar en agregados.

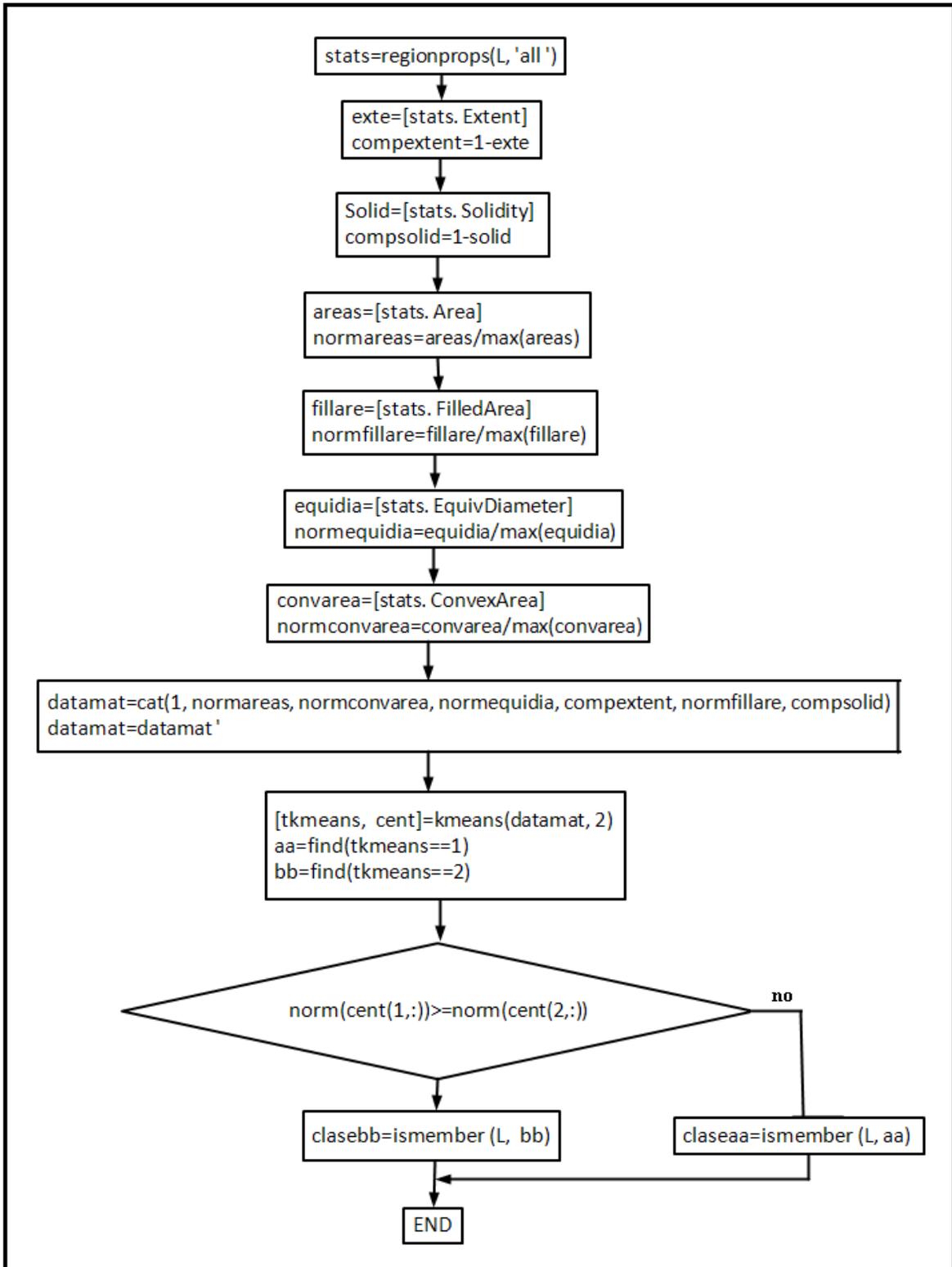


Figura 2. 10: Diagrama de flujo del algoritmo *K-medias* para separar en dos clases: agregados y no agregados.

2.6 Segmentación de agregados empleando la transformada *watershed*

Una vez que se definen cuales son los agregados se procede a separar las células que forman los mismos, los pasos seguidos para la separación de los agregados son:

1. Se obtiene el negativo de la transformada de distancia de la máscara binaria (imagen que contiene los agregados) empleando la distancia euclidiana.
2. Se normaliza y calcula el inverso a la transformada de distancia, ver Figura 2.11 (a) y (b).

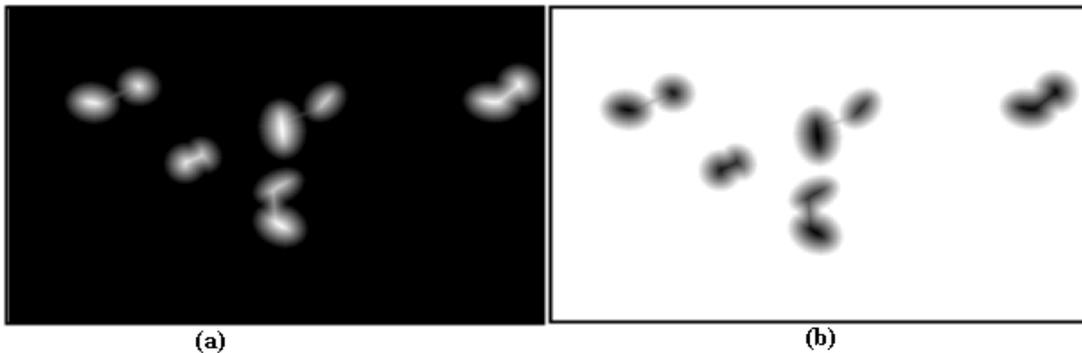


Figura 2.11: (a) Transformada de distancia normalizada. (b) Transformada de distancia inversa.

3. Se determinan los centroides de las regiones expandidas, los cuales son los mínimos que identifican cada célula y se crea una máscara con los mismos. Las regiones expandidas y sus centroides serán en este caso los marcadores internos y el fondo de la imagen actúa como marcador externo.
4. A la transformada de distancia inversa se le aplica la función *imimposemin* de Matlab, para imponer mínimos en la posición de los marcadores internos que se van a emplear y eliminar cualquier otro mínimo que pueda existir en la imagen de la transformada de distancia inversa, ver Figura 2.12.

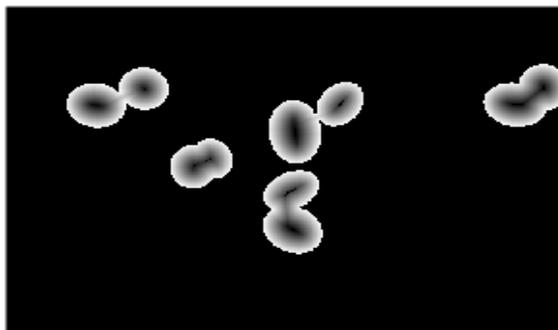


Figura 2.12: Imposición de mínimos.

5. Se calcula la transformada *watershed* como se muestra en la Figura 2.13.

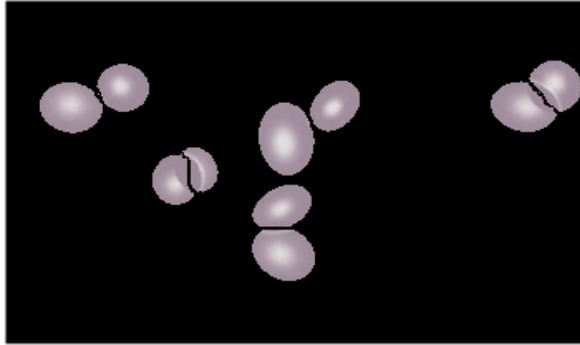


Figura 2.13: Resultado final de la segmentación.

6. Una vez realizado este procedimiento de separación, se procede a reconstruir la máscara original.

Los pasos explicados anteriormente se encuentran en el anexo (1).

CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

En este capítulo se hace un análisis entre los algoritmos implementados para la detección de agregados y se realiza una comparación estadística entre los mismos. También se demuestran los resultados de la segmentación obtenidos con el método de Otsu y el algoritmo de Lloyd, que sirven como datos de partida. Además se muestran los resultados obtenidos en el proceso de segmentación fina de los agregados que fue implementado.

3.1 Resultados de la segmentación con el método de Otsu y el algoritmo de Lloyd

La segmentación por el método de Otsu, utilizando la componente de intensidad (V) en el espacio HSV, de imágenes (en este caso de eritrocitos), produce resultados como los que se muestran en la Figura 3.1. Cuando la morfología de los mismos es poco compleja, estas imágenes pueden considerarse cualitativamente adecuadas en cuanto a la posibilidad de visualizar en ellas los agregados y distinguirlos de las células aisladas.

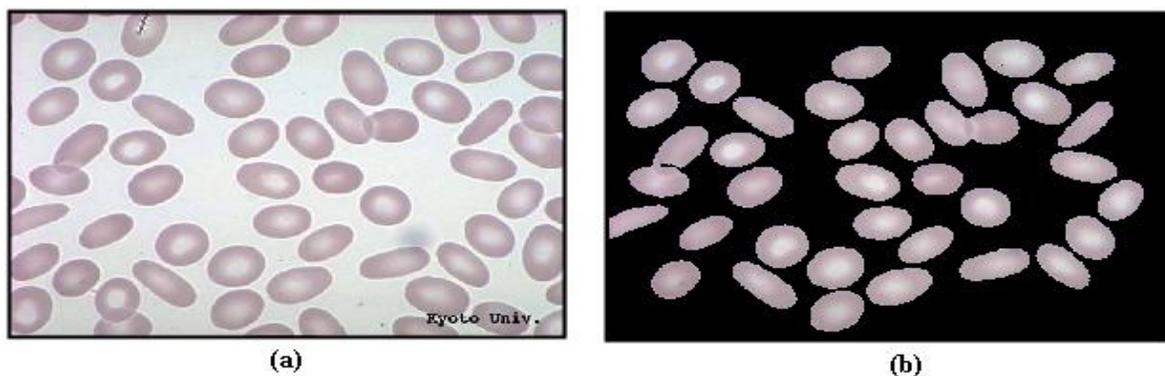


Figura 3. 1: (a) Imagen "Elliptocyte". (b) Imagen "Elliptocyte" segmentada por el método de Otsu.

La segmentación por el método de Lloyd también produce resultados satisfactorios para los propósitos de este trabajo, como se puede observar en la Figura 3.2. Los resultados de esta

segmentación sirvieron como imágenes de partida para la detección automática de los agregados contenidos en las imágenes correspondientes.

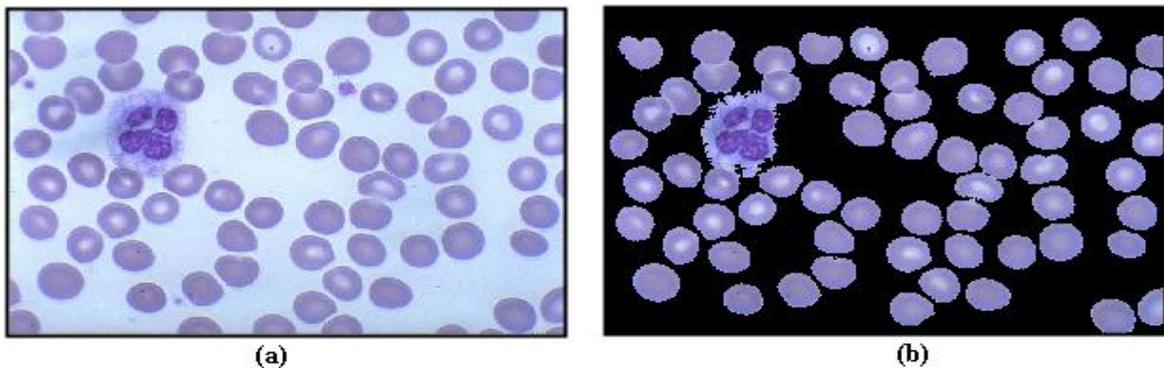


Figura 3. 2: (a) Imagen original ‘img04’. (b) Imagen ‘img04’ segmentada por el método de Lloyd.

3.2 Resultados y comparación entre los experimentos propuestos para la detección de agregados

En los experimentos realizados el primero (Detección de agregados utilizando la transformada de distancia y la transformada *H-maxima*) tuvo mejores resultados en cuanto a la identificación de los agregados con respecto al segundo experimento (Algoritmo *K-medias* para separar en 2 clases: agregados y no agregados). Se tomó como ejemplo la imagen ‘Imagen 65’ que se muestra en la Figura 3.3.

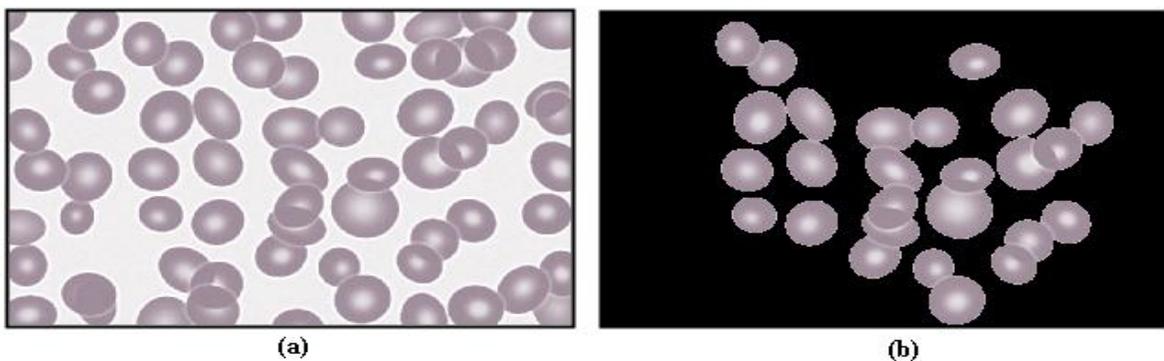


Figura 3. 3: (a) ‘Imagen 65’. (b) Segmentación gruesa de la ‘Imagen 65’.

En la Figura 3.4 (a) se muestran los resultados del primer experimento donde se observan todos los agregados correctamente identificados, por el contrario en la Figura 3.4 (b) y (c) se muestra la detección de agregados y no agregados respectivamente del segundo experimento señalando en color amarillo los agregados detectados incorrectamente.

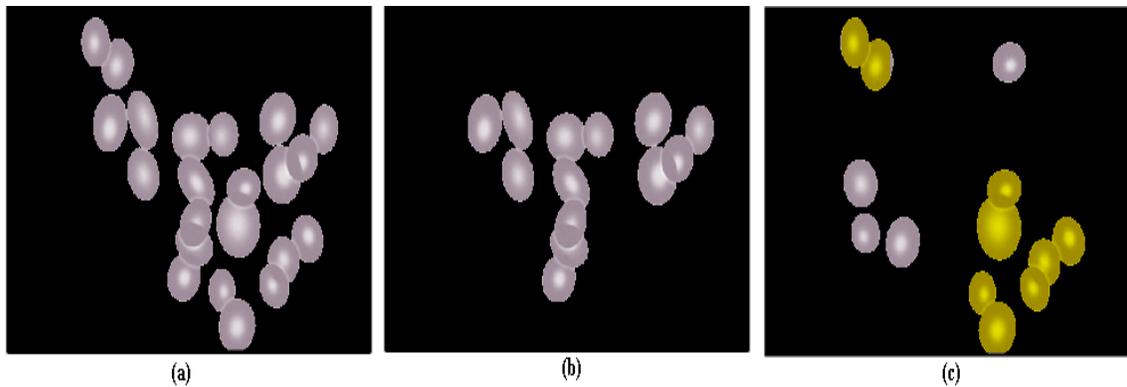


Figura 3.4: (a) Primer experimento con la imagen “Imagen 65”. (b) Segundo experimento con la imagen “Imagen 65” (primera clase: agregados). (c) Segundo experimento con la imagen “Imagen 65” (segunda clase: no agregados).

Estos resultados también se pueden observar en las Tabla 3.1 y Tabla 3.2 donde se encuentran todos los nombres de las imágenes que se utilizaron en los experimentos incluida la que se tomó como ejemplo, donde se exponen el número total de objetos en cada imagen al igual que la cantidad de verdaderos positivos (cantidad de objetos aglomerados detectados correctamente), falsos positivos (cantidad de objetos aislados detectados como aglomerados), falsos negativos (cantidad de objetos aglomerados detectados como objetos aislados), verdaderos negativos (cantidad de objetos aislados detectados correctamente), la sensibilidad (es la probabilidad de que los objetos aglomerados sean detectados correctamente) y especificidad (es la probabilidad de que los objetos aislados sean detectados correctamente) para cada una de las imágenes.

Tabla 3.1: Resultados del primer experimento para detectar aglomerados (Detección de agregados utilizando la transformada de distancia y la transformada H -maxima). La segmentación gruesa fue con el método de Otsu.

Nombre de la imagen	No. de objetos	VP	FP	FN	VN	Sensibilidad	Especificidad
						(Se)	(Sp)
						$\frac{VP}{VP + FN}$	$\frac{VN}{VN + FP}$
Imagen	17	5	0	0	12	1	1
Imagen2	18	3	0	1	14	0.75	1
Imagen65	11	7	0	0	4	1	1
Eritrocitos sintéticos	31	12	1	2	16	0.85	0.94
Elliptocyte	35	2	0	0	33	1	1
Imagen4	58	1	2	0	55	1	0.96
Im04	55	5	4	0	46	1	0.92
Pappenheimers body	28	1	8	0	19	1	0.70
Rouleaux formation	7	3	0	0	4	1	1
Spherocyte	24	2	3	0	19	1	0.86
Tear Drop Cell	32	1	4	0	27	1	0.87
Imag1	42	24	0	4	14	0.85	1
Imag2	17	3	0	0	14	1	1
Imag3	20	8	1	0	11	1	0.92
Imag4	23	3	4	0	16	1	0.80
Imag5	22	7	1	0	14	1	0.93
Imag6	32	3	7	0	22	1	0.76
Imag7	126	28	4	0	94	1	0.96
Imag8	15	5	3	0	7	1	0.70
Imag9	19	2	2	3	12	0.40	0.85
Imag10	191	35	0	0	156	1	1
Imag11	28	7	0	0	21	1	1
Imag12	25	9	4	1	11	0.90	0.73
Imag13	27	1	6	0	20	1	0.77

Tabla 3.2: Resultados del segundo experimento para detectar aglomerados (Algoritmo *K-medias* para separar en 2 clases: agregados y no agregados). La segmentación gruesa fue con el método de Otsu.

Nombre de la imagen	No. de objetos	VP	FP	FN	VN	Sensibilidad	Especificidad
						(Se)	(Sp)
						$\frac{VP}{VP + FN}$	$\frac{VN}{VN + FP}$
Imagen	17	4	0	1	12	0.80	1
Imagen2	18	3	0	1	14	0.75	1
Imagen65	11	3	0	4	4	0.42	1
Eritrocitos sintéticos	30	2	0	11	17	0.15	1
Elloptocyte	35	2	0	0	33	1	1
Imagen4	58	1	8	0	49	1	0.85
Im04	55	5	0	0	50	1	1
Pappenheimers body	28	1	16	0	12	1	0.42
Rouleaux formation	7	1	0	2	4	0.33	1
Spherocyte	24	2	3	0	19	1	0.86
Tear Drop Cell	32	1	4	0	27	1	0.87
Imag1	42	7	0	21	14	0.85	1
Imag2	17	3	1	0	13	1	0.93
Imag3	20	3	0	5	12	0.38	1
Imag4	23	3	0	0	20	1	1
Imag5	22	2	0	5	15	0.29	1
Imag6	32	3	23	0	6	1	0.21
Imag7	126	26	1	2	97	0.93	0.99
Imag8	15	3	0	2	10	0.60	1
Imag9	19	4	0	1	14	0.80	1
Imag10	191	28	3	7	153	0.80	0.98
Imag11	28	6	0	1	21	0.86	1
Imag12	25	10	0	0	15	1	1
Imag13	27	1	3	0	23	1	0.88

3.3 Validación estadística entre los experimentos propuestos para la detección de agregados

En los experimentos realizados el primero (Detección de agregados utilizando la transformada de distancia y la transformada *H-maxima*) tuvo mejores resultados en cuanto a la identificación de los agregados con respecto al segundo experimento (Algoritmo *K-medias* para separar en 2 clases: agregados y no agregados). En la prueba de rangos con signo de *Wilcoxon*, se tiene la hipótesis nula consistente en considerar que las medianas de los valores obtenidos para la sensibilidad y la especificidad por los dos métodos no difieren. Este análisis estadístico se realizó mediante el software SPSS 11.5, a través de una prueba no paramétrica para muestras relacionadas.

En este caso, se obtuvieron los resultados que se muestran en las tablas del anexo II realizadas para la sensibilidad y la especificidad por los dos métodos implementados con un nivel de significación de 0.05. Se obtuvo que la hipótesis nula se rechaza para la sensibilidad ya que el valor de la probabilidad en el punto es 0.016 como se puede observar en la Figura 3.5 (a), mientras que en el caso de la especificidad la hipótesis nula se acepta al ser el valor de la probabilidad en el punto de 0.092, ver Figura 3.5 (b). También en la Figura 3.6 (a) se observa que en cuanto a la sensibilidad hay diferencias significativas entre los métodos y que fue el primero el que mayor valor obtuvo para la media y para la mediana. En la especificidad el primer método y el segundo método no tienen diferencias significativas, esto se puede observar en las medias donde existen similares valores de variabilidad como se muestra en la Figura 3.6 (b). Entonces, no tenemos razones confiables para asegurar que la especificidad se diferencie entre uno u otro método.

Estadísticos de contraste ^c	
	Sensibilidad Método 2 - Sensibilidad Método 1
Sig. exacta (bilateral)	.039 ^{a,b}
Sig. exacta (unilateral)	.019 ^b
Probabilidad en el punto	.016 ^b
a. Se ha usado la distribución binomial. b. Se ofrecen los resultados exactos en vez de los de Monte Carlo para esta prueba. c. Prueba de los signos	

(a)

Estadísticos de contraste ^c	
	Especificidad Método 2 - Especificidad Método 1
Sig. exacta (bilateral)	.302 ^{a,b}
Sig. exacta (unilateral)	.151 ^b
Probabilidad en el punto	.092 ^b
a. Se ha usado la distribución binomial. b. Se ofrecen los resultados exactos en vez de los de Monte Carlo para esta prueba. c. Prueba de los signos	

(b)

Figura 3.5: (a) Estadísticos de contraste de la sensibilidad. (b) Estadísticos de contraste de la especificidad.

Estadísticos			
		Sensibilidad Método 1	Sensibilidad Método 2
N	Válidos	24	24
	Perdidos	0	0
Media		.9479	.7900
Mediana		1.0000	.8950
Varianza		.01793	.07510

(a)

Estadísticos			
		Especificidad Método 1	Especificidad Método 2
N	Válidos	24	24
	Perdidos	0	0
Media		.9029	.9163
Mediana		.9350	1.0000
Varianza		.01133	.03787

(b)

Figura 3. 6: (a) Estadísticos de la sensibilidad. (b) Estadísticos de la especificidad.

3.4 Segmentación fina de los eritrocitos aglomerados

En este trabajo las aglomeraciones fueron detectadas experimentalmente siguiendo la metodología que se expuso en el capítulo 2. Existen diferentes casos de aglomeraciones, que pueden ser de dos o más eritrocitos lo cual se observa en la Figura 3.7 .

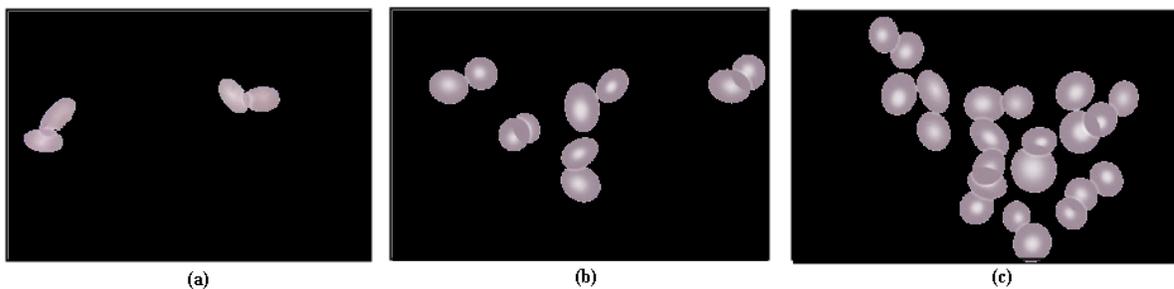


Figura 3.7: (a) Aglomeraciones detectadas en la imagen “*Elliptocyte*”. (b) Aglomeraciones detectadas en la imagen “*Imagen*”. (c) Aglomeraciones detectadas en la imagen “*Imagen 65*”.

En la Figura 3.8 se muestra cómo fueron separadas las aglomeraciones mediante la transformada *watershed* con imposición de mínimos para aliviar el posible problema de la sobre-segmentación, con lo que se obtuvieron resultados satisfactorios desde el punto de vista cualitativo ya que se observó que los aglomerados fueron subdivididos en sus partes componentes. La precisión de esta segmentación fina deberá ser evaluada cuantitativamente en trabajos futuros.

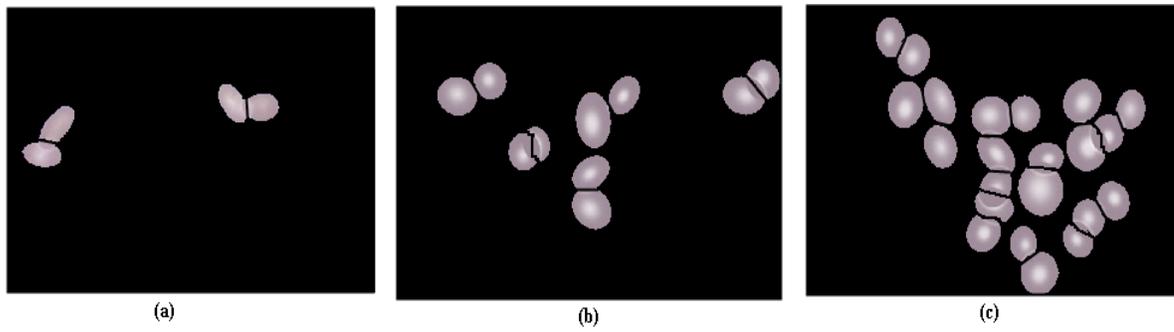


Figura 3. 8: (a) Imagen ‘*Elliptocyte*’ con los aglomerados segmentados. (b) Imagen ‘Imagen’ con los aglomerados segmentados. (c) Imagen ‘Imagen 65’ con los aglomerados segmentados.

CONCLUSIONES

1. En el primer método de detección de agregados utilizando operaciones morfológicas (transformada de distancia y transformada *H-maxima*) se obtuvieron buenos resultados en cuanto a la sensibilidad con un 94% y la especificidad con un 78%.
2. En el segundo método de detección de agregados utilizando el algoritmo *K-medias* no se obtuvieron buenos resultados en cuanto a la sensibilidad con un 65% y la especificidad con un 91%.
3. La comparación en cuanto al desempeño de los dos métodos reveló que el primer método fue superior al segundo método al aplicarle la prueba de los rangos con signos *Wilcoxon*. El segundo método es dependiente de los rasgos que hayan sido escogidos, mientras que el primer método emplea características morfológicas de las mascararas binarias resultantes de la segmentación de los objetos a clasificar.
4. La segmentación fina de las aglomeraciones por el algoritmo de la transformada *watershed* con imposición de mínimos fue implementada con resultados satisfactorios.

RECOMENDACIONES

1. Estudiar otros descriptores de rasgos para mejorar el método de detección de aglomerados basado en el algoritmo *K-medias* y así aumentar sus potencialidades y comparar los nuevos resultados con el algoritmo basado en la transformada *H-maxima* y la transformada de distancia.
2. Seguir perfeccionando los métodos de detección de agregados para permitir un mayor nivel de automatización en el proceso general de segmentación.
3. Evaluar cuantitativamente los resultados de la segmentación de los agregados e introducir otros métodos de segmentación de estos, que permitan mejorar los resultados obtenidos a partir de los métodos empleados, como por ejemplo utilizando la posibilidad de aprovechar la información que contiene la imagen acerca de la verdadera frontera entre las células que se solapan.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Nasr-Isfahani, S., A. Mirsafian, et al. (2008). A New Approach for Touching Cells Segmentation. International Conference on BioMedical Engineering and Informatics, 2008. BMEI 2008.
- [2] Cloppet, F. and A. Boucher (2008). Segmentation of overlapping/aggregating nuclei cells in biological images. 19th International Conference on Pattern Recognition, 2008. ICPR 2008.
- [3] Zhang, S., Q. Jinlong, et al. (2008). Urinary Sediment Overlapping Cells Image Segmentation Based on Combination Strategy. International Symposium on Computational Intelligence and Design, 2008. ISCID '08.
- [4] Sheehy, A., G. Martinez, et al. (2008). Region and contour based cell cluster segmentation algorithm for in-situ microscopy. Electrical Engineering, 5th International Conference on Computing Science and Automatic Control, 2008. CCE 2008.
- [5] Ali, R., M. Gooding, et al. (2007). Phase-based segmentation of cells from brightfield microscopy. 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007.
- [6] Gonzalez, R. C. and Woods, R. E. [2002]. *Digital Image Processing*, 2nd ed, Prentice Hall, Upper Saddle River, NJ.
- [7] Cheolha Pedro Lee, [2005]. Robust Image Segmentation using Active Contours: Level Set Approaches. Dept. of Electrical and Computer Engineering North Carolina State University.

- [8] Gutiérrez Hernández Aristeo, julio de [2008]. Conectividad difusa y segmentación de imágenes digitales. Universidad tecnológica de la Mixteca, Huajuapán de León, Oaxaca.
- [9] Lloyd, S. P. (1982). "Least squares quantization in PCM", *IEEE Transactions on Information Theory*, 28(Marzo): 129-137.
- [10] Gerscho A. y R. M. Gray (1996). *Vector quantization and signal compression*. 5ta ed, Kluwer Academic Publishers, pp 188-189, Estados Unidos.
- [11] Medeiros, N. (2010). *Atlas of hematology*, disponible en:
<http://www.hematologyatlas.com/>, accedido el 10 /02/2010.
- [12] Ichihashi T. (12/1996). *Atlas of Hematology*, disponible en:
<http://pathy.met.nagoyau.ac.jp/atlas/doc/>, accedido el 24 /11/2010.
- [13] Russ, J. C. (1999). *The image processing handbook*. 3ra. ed, CRC Press, Raleigh, North Carolina.
- [14] Pratt, W. K. (2001). *Digital image processing*. 3ra. ed, Wiley-Interscience, pp 45-87,241-270, Los Altos, California.
- [15] Díaz, G., Fabio González, Eduardo Romero (2008). Automatic Clump Splitting for Cell Quantification in Microscopical Images Progress in Pattern Recognition, Image Analysis and Applications Lecture Notes in Computer Science **4756/2008**: 763-772.
- [16] Mao, K. Z., P. Zhao, et al. (2003). Overlapping/touching cell nuclei segmentation based on analysis of perpendicular distance curve. Biomedical Engineering, 2003. IEEE EMBS Asian-Pacific Conference on.
- [17] Quan, P., Y. Cuirong, et al. (2006). Overlapped Cell Image Segmentation Based on Distance Transform. Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on.
- [18] Wang, W. and H. Song (2007). Cell Cluster Image Segmentation on Form Analysis. Natural Computation, 2007. ICNC 2007. Third International Conference on.
- [19] Yan, W. and X. Feng (2010). A Watershed Based Segmentation Method for Overlapping Chromosome Images. Education Technology and Computer Science (ETCS), 2010 Second International Workshop on.
- [20] Bergen, T., Steckhan, Dirk, Wittenberg, Thomas, Zerfass, Thorsten (2008). Segmentation of leukocytes and erythrocytes in blood smear images. Engineering in

- Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE.
- [21] Adollah, R., M. Y. Mashor, N. F. Mohd Nasir, H. Rosline, H. Mahsin and H. Adilah (2008). Blood Cell Image Segmentation: A Review 4th Kuala Lumpur International Conference on Biomedical Engineering 2008. S. B. Heidelberg. Kuala Lumpur. **Volume 21:** 141-144.
- [22] Besbes, M. (2004). Cells images color segmentation based on thresholding and watershed segmentation. Industrial Technology, 2004. IEEE ICIT '04. 2004 IEEE International Conference on.
- [23] Espinoza, E., G. Martínez, et al. (2006). Cell cluster segmentation based on global and local thresholding for in-situ microscopy. Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on.
- [24] Gómez-Gil, P., M. Ramírez-Cortes, et al. (2008). A Feature Extraction Method Based on Morphological Operators for Automatic Classification of Leukocytes. Artificial Intelligence, 2008. MICAI '08. Seventh Mexican International Conference on.
- [25] Ali, R., M. Gooding, et al. (2008). Advanced phase-based segmentation of multiple cells from brightfield microscopy images. Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on.
- [26] Angulo, J. and B. Schaack (2008). Morphological-based adaptive segmentation and quantification of cell assays in high content screening. Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on.
- [27] González RC, Woods RE, Eddins SL, "Digital Image Processing Using Matlab", Prentice-Hall, NJ, 2004.
- [28] Plataniotis, K. N. y A. N. Venetsanopoulos (2000). *Color image processing and applications*. Springer-Verlag, pp 25-31,266-268.
- [29] Irwin R. Miller, John E. Freund y Richard Johnson (1992). *Probabilidad y estadística para ingenieros*, Prentice-Hall Hispanoamericana, S.A, tomo 2.

ANEXOS

A continuación se muestran los anexos con los programas implementados y las tablas utilizadas en este trabajo.

Anexo I Programas utilizados

General Otsu: programa para segmentar imágenes mediante el algoritmo de Otsu

%Se lee la imagen.

```
rgb_image=imread ('D:\Tesis\Imágenes\Imagen.bmp');
```

%Se convierte del espacio rgb al hsv.

```
hsv_image = rgb2hsv (rgb_image);
```

```
v= hsv_image (:,:,3);
```

%Se aplica el método de Otsu, (segmentación gruesa).

```
level= graythresh(v);
```

%Se binariza la imagen.

```
BW = im2bw (v,level);
```

%Se halla el inverso de la imagen.

```
BW2=~BW;
```

%Se recortan los bordes de la imagen.

```
IM2 = imclearborder(BW2);
```

%Se rellenan los "huecos" de la imagen.

```
recons = imfill(IM2,'holes');  
  
%Se eliminan los objetos extraños (pequeños) en la imagen.  
  
recons = bwareaopen(recons,1000);  
  
%Se etiqueta la imagen.  
  
[L, NUM]= bwlabel(recons);
```

General Lloyd: programa para segmentar imágenes mediante el algoritmo de Lloyd

```
% Se lee la imagen.  
  
Original =imread ('D:\Tesis\Imágenes\Imagen.bmp');  
  
imagen=im2double(Original);  
  
rojo=imagen(:,:,1);  
  
verde=imagen(:,:,2);  
  
azul=imagen(:,:,3);  
  
% Obtener los valores de los planos de color por columnas  
  
matrizRGB=[rojo(:)';verde(:)';azul(:)'];  
  
% Se le aplica a la matriz matrizRGB, el cálculo del valor promedio, donde se obtiene un  
vector de tres valores, cada uno correspondiente a un plano de color; primero rojo, luego  
verde y posteriormente azul.  
  
prom=mean(matrizRGB);  
  
[rows columns height]=size(imagen);  
  
% Creo el espacio para las matrices máscaras.  
  
matrizpos=zeros([rows columns]);  
  
mascaranegat=zeros([rows columns]);  
  
% Le sumo y le resto 0.1, para obtener dos nuevos vectores con los cuales se compara,  
para crear la máscara.  
  
promneg=prom-0.1;
```

```
prompos=prom+0.1;
promini=0;
promfin=inf;
n=0;
% Se realiza un while que para cuando la diferencia entre las distancias es muy pequeña.
while (abs(promfin-promini)>0.1 )
%La función nl3filter me permite un barrido por toda la imagen, píxel a píxel y en cada
pixel se calcula la distancia euclidiana con la función distanceLya. Se define el tamaño de
la ventana de 1x1, y se le pasa como parámetro de la función distanceLya, el vector
promnegat y prompos, en dependencia del cálculo a realizar.
Distneg= nl3filter(imagen,[1 1],'distanceLya',promneg);
Distpos = nl3filter(imagen,[1 1],'distanceLya',prompos);
Distneg=Distneg(:,:,1);
Distpos=Distpos(:,:,1);
% Promini es una variable que toma el valor del promedio de las distancias para poder
compararlo con el nuevo valor de distancia para obtener un criterio de parada que seria
cuando una poca variación de un valor a otro.
promini=promfin;
% Resto las matrices de distancia, para obtener una nueva matriz diferencia, donde los
valores positivos corresponderían a los más cercanos a Distpos y los negativos serían los
más cercanos a Distneg. En este caso se toma un plano porque el nl3filter, repite en los tres
planos el resultado, ya que se basa en la dimensión de la matriz que analiza.
matrizdis=Distneg-Distpos;
% Busco los píxeles positivos para formar una máscara.
matrizpos(find(matrizdis<=0))=1;
% Pongo los valores más cercanos a prompos en cero.
```

```
matrizpos=(~matrizpos);  
  
% Obtengo una máscara 3D.  
  
mascarapos=cat(3,matrizpos,matrizpos,matrizpos);  
  
% Multiplico la máscara por la imagen, para obtener la nueva imagen, donde los valores  
más cercanos al vector prompos serían cero.  
  
imagmascpos=imagen.*mascarapos;  
  
imagmascpos_red=imagmascpos(:, :, 1);  
  
imagmascpos_green=imagmascpos(:, :, 2);  
  
imagmascpos_blue=imagmascpos(:, :, 3);  
  
posred=imagmascpos_red(find(imagmascpos_red~=0));  
  
posgreen=imagmascpos_green(find(imagmascpos_green~=0));  
  
posblue=imagmascpos_blue(find(imagmascpos_blue~=0));  
  
posmeanred=mean(posred);  
  
posmeangreen=mean(posgreen);  
  
posmeanblue=mean(posblue);  
  
% Obtengo un nuevo vector, cuyos valores se corresponden al promneg, ya que  
anteriormente con la máscara se hizo cero los valores más cercanos a prompos.  
  
promneg=[posmeanred posmeangreen posmeanblue];  
  
% Se obtiene la máscara, pero en este caso la más cercana a promneg. Se realiza la misma  
operación que en caso anterior.  
  
mascaraneg=(~mascarapos);  
  
imagmascneg=imagen.*mascaraneg;  
  
imagmascneg_red=imagmascneg(:, :, 1);  
  
imagmascneg_green=imagmascneg(:, :, 2);  
  
imagmascneg_blue=imagmascneg(:, :, 3);
```

```
negred=imagmascneg_red(find(imagmascneg_red~=0));
neggreen=imagmascneg_green(find(imagmascneg_green~=0));
negblue=imagmascneg_blue(find(imagmascneg_blue~=0));

negmeanred=mean(negred);

negmeangreen=mean(neggreen);

negmeanblue=mean(negblue);

prompos=[negmeanred negmeangreen negmeanblue];

% Para el criterio de parada de la función while obtengo el promedio de todos los valores
de las dos matrices de distancia, y lo llamo promfin.

maskn=mascaraneg(:,:,1);
maskp=mascarapos(:,:,1);
restneg=Distneg.*maskn;
restpos=Distpos.*maskp;
Dist= restneg+ restpos;
promfin=mean(Dist(:));

n=n+1;

end

maskneg=mascaraneg(:,:,1);
maskpos=mascarapos(:,:,1);

% Para recortar los bordes de la máscara positiva
maskpos=imclearborder(maskpos);

% Para recortar los bordes de la máscara negativa
maskneg==imclearborder(maskneg);

[Lpos cantpos] = bwlabel(maskpos);
[Lneg cantneg] = bwlabel(maskneg);
```

```
if cantneg>cantpos
    mask=maskpos;
    % Para rellenar huecos de la máscara positiva.
    mask2 =imfill (mask ,'holes');
    % Se eliminan los puntos extraños
    mask3 = bwareaopen(mask2 ,1000);
    mask4=cat(3,mask3,mask3,mask3);
    segmcolor=imagen.*mask4;
    mascara=mask4(:,:,1);
    [L, NUM] = bwlabel(mascara);
end
if cantneg<cantpos
    mask=maskpos;
    mask5 =imfill (mask ,'holes');
    mask6 = bwareaopen(mask5 ,1000);
    mask7=cat(3,mask6,mask6,mask6);
    segmcolor=imagen.*mask7;
    mascara2=mask7(:,:,1);
    [L, NUM] = bwlabel(mascara2);
end
```

Primer método de detección de agregados utilizando la transformada de distancia y la transformada *H-maxima*

```
% Matriz que contendrá los agregados.
aglom = zeros(size(L));
% Matriz que contendrá los máximos.
```

```
ptomax = zeros(size(L));

% Se inicia el contador cont.

cont=0;

% Se inicia el contador h.

h=1;

% Se inicia el umbral j.

j=0;

% El valor del incremento de la j.

dj=0.001;

% Vector de 10 elementos (todos 0) donde se guardarán la cantidad de subregiones para
comparar.

A=[0];

for i = 1 :NUM

% Se va de región en región.

BW1 = (L == i);

% Se calcula la transformada de distancia.

D = bwdist(~BW1);

% Se normaliza.

D2 = D/max(max(D));

% Se calcula la transformada h-máxima extendida.

D3=imextendedmax(D2,j);

%Se enumeran las subregiones.

[DL, DNUM] = bwlabel(D3);

if DNUM >1

while cont<=5;
```

```
j=j+dj;
% Se calcula la transformada h-máxima extendida.
D4=imextendedmax(D2,j);
% Se etiqueta para comprobar la cantidad de subregiones.
[DL4, DNUM4] = bwlabel(D4);
if DNUM4==A(h);
cont=cont+1;
else cont=0;
end
A=[A DNUM4];
h=h+1;
end
if A(h-1)>1
% Se hace un OR donde se adicionan a la matriz aglom los nuevos aglomerados
detectados.
aglom= aglom | BW1;
%Se hace un OR donde se adicionan a la matriz ptomax los puntos máximos de los
aglomerados detectados.
ptomax= ptomax | D4;
end
% Se limpia el contador.
cont=0;
%Se limpia el vector.
A=[0];
% Se inicia el contador h nuevamente.
```

```
h=1;
end
end
X=L.*aglom;
X2=L.*ptomax;
```

Segundo método: Algoritmo *K-means* para separar en 2 clases: agregados y no agregados

```
stats = regionprops(L,'all');
% Se calcula las áreas
areas=[stats.Area];
normareas=areas/max(areas);
% Se calcula el área convexa
convarea=[stats.ConvexArea];
normconvarea=convarea/max(convarea);
% Se calcula EquivDiameter (diámetro equivalente)
equidia=[stats.EquivDiameter];
normequidia=equidia/max(equidia);
% Se calcula Extent (extensión)
exte=[stats.Extent];
compextent=1-exte;
% Se calcula FilledArea
fillare=[stats.FilledArea ];
normfillare=fillare/max(fillare);
% Se calcula la solides
solid=[stats.Solidity];
```

```
compsolid=1-solid;

% Matriz para concatenar todos los rasgos.

datamat=cat(1,normareas,normconvarea,normequidia,compeitent,normfillare,
compsolid);

datamat=datamat';

% Se divide en 2 clases.

[Tkmeans,cent]=kmeans(datamat,2);

aa=find(Tkmeans==1);

bb=find(Tkmeans==2);

if norm(cent(1,:))>=norm(cent(2,:))

clasebb=ismember(L,bb);

else

claseaa=ismember(L,aa);

end
```

Programa para calcular los centroides de los puntos máximos de los agregados detectados

```
%Se etiquetan los máximos.

[DL5,DNUM5]=bwlabel(X2);

%Se calculan los centrides.

stats = regionprops(DL5,'Centroid');

Centroide=[stats.Centroid];

%Se normalizan los centroides

NormaliCentroide = Centroide/max(max(Centroide));

% Se halla la traspuesta.

centro=NormaliCentroide';
```

% Se hace una matriz de 2 por el número de centroides.

```
yy=flipud(round(reshape(centro,2,DNUM5)));
```

% Se iguala a mask la imagen de los máximos.

```
mask = X2;
```

% Se pone en negro la región de interés.

```
marker = false(size(mask));
```

```
Z=logical(yy);
```

% Se pone en blanco la región de interés.

```
marker(Z) == true;
```

```
J = mask;
```

```
J(marker) = 255;
```

Programa para realizar la segmentación fina con la transformada *watershed* y la imposición de mínimos

% Se calcula la transformada de distancia de los aglomerados detectados.

```
dist=bwdist(~X);
```

% Se normaliza la transformada de distancia.

```
normdist = dist/max(max(dist));
```

% Se calcula el inverso de la transformada de distancia.

```
inver=1-normdist;
```

% La posición de los centroides.

```
im=J;
```

% Se pone en negro la región de interés.

```
em=normdist==0;
```

% Se imponen los mínimos.

```
imposi=imimposemin(inver ,im|em);
```

```
% Se calcula la transf. de watershed.  
transwater=watershed(imposi);  
water=X;  
water(transwater==0)=255;  
  
% Se realiza un AND para mostrar la segmentación.  
resulwater=transwater&water;
```

Función para calcular la distancia euclidiana entre los dos vectores de entrada.

```
function eucl=distanceLya(vectpix,vectprom);  
  
% El vector vectpix, se tiene que llevar a una dimensión puesto que de la  
% función nl3filter obtengo un vector tridimensional.  
vectpix=vectpix(1,:);  
  
% Resto los vectores y obtengo un vector diferencia.  
vdif=vectpix-vectprom;  
  
% La función norm calcula la distancia euclidiana en el vector vdif.  
eucl=norm(vdif);  
  
% size(eucl)
```

Anexo II Tablas utilizadas

Tabla 1: Prueba de los rangos con signos de *Wilcoxon* para la sensibilidad.

Prueba de los rangos con signo de *Wilcoxon*

Rangos				
		N	Rango promedio	Suma de rangos
Sensibilidad Método 2 -	Rangos negativos	10 ^a	6.95	69.50
Sensibilidad Método 1	Rangos positivos	2 ^b	4.25	8.50
	Empates	12 ^c		
	Total	24		

a. Sensibilidad Método 2 < Sensibilidad Método 1
 b. Sensibilidad Método 2 > Sensibilidad Método 1
 c. Sensibilidad Método 2 = Sensibilidad Método 1

Tabla 2: Prueba de los rangos con signos de *Wilcoxon* para la especificidad.

Prueba de los rangos con signo de *Wilcoxon*

Rangos				
		N	Rango promedio	Suma de rangos
Especificidad Método 2 -	Rangos negativos	5 ^a	8.40	42.00
Especificidad Método 1	Rangos positivos	10 ^b	7.80	78.00
	Empates	9 ^c		
	Total	24		

a. Especificidad Método 2 < Especificidad Método 1
 b. Especificidad Método 2 > Especificidad Método 1
 c. Especificidad Método 2 = Especificidad Método 1