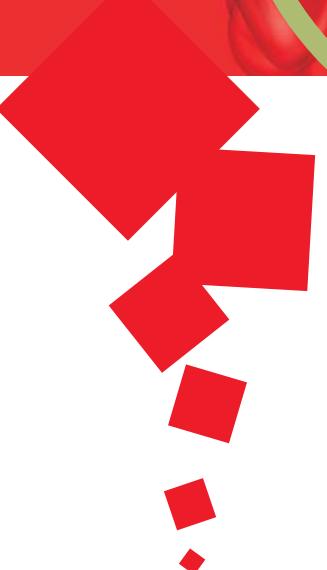




UNIVERSIDAD DE ORIENTE

FACULTAD MATEMÁTICA-COMPUTACIÓN

TESIS EN OPCIÓN AL TÍTULO DE LICENCIADO EN CIENCIA DE LA COMPUTACIÓN



AUTORES:

**GRETHEL YUSEL COELLO SAID
PEDRO DIAMEL MARRERO FERNÁNDEZ**

TUTORES:

**MSC. SILENA HEROLD GARCÍA
MSC. MIRIELA ESCOBEDO NICOT
DRA. KARELIS FERNÁNDEZ ARIAS**

A Joiker, Dorita

-Grethel Coello

A Albertina

-Pedro Marrero

AGRADECIMIENTOS.

Quiero agradecer profundamente a todas las personas que colaboraron de una forma u otra en la realización de este trabajo. A mi compañera de trabajo Grethel por haber compartido conmigo estos meses, día y noche. Quiero agradecer también a mis tutoras Silena Herold, Miriela Escobedo y Karelis Fernández por haberme guiado, aconsejado y haberme soportado durante todo el desarrollo del trabajo. De manera especial a mi tío Ibrahim por ser un ejemplo para mí y a mi tío Pedro por su ayuda incondicional, ambos han sido como padres para mí, a Roberto por sus sabios consejos y valoraciones. En el plano personal quisiera agradecer a Dorita, Joiker, mi tía Soraya y mis padres por ser un apoyo espiritual y material que nunca olvidaré. Y agradecer por último a mi abuela Albertina la persona a quien le debo lo que soy.

-Pedro Marrero.

Merecen un especial agradecimiento mis tutoras Silena Herold, Miriela Escobedo y Karelis Fernández por sus oportunas y sabias orientaciones, por transmitirme sus conocimientos y experiencias. A mi mamá por su apoyo ilimitado, sacrificio, volcada en función de mis necesidades y tolerante ante mis debilidades emocionales. En especial a mi compañero Pedro por su comprensión, entrega total a lo largo de mi carrera, y su paciencia sobre todo en estos meses. Debo reconocer a mi amigo Robertico por su ayuda sincera y constante desde los inicios de mi carrera. También a Ibrahim por estar siempre dispuesto a enseñarme y colaborar en la confección del trabajo. A Naty, Carmen, Albertina, Soraya que en los momentos precisos me sirvieron de sostén. A la Dra. Salmon por su colaboración, y a todos los que contribuyeron en la realización de la tesis. Por último y más importante a mi hermano Joiker por haberme dado la oportunidad de estudiar mi carrera, por su amor, por su sacrificio.

-Grethel Coello.

Es justamente la posibilidad de realizar un sueño lo que hace que la vida sea interesante.

-Paulo Coelho

RESUMEN.

La deformación de las células en la sangre es el primer exponente de la existencia de ciertos padecimientos o enfermedades. El conocimiento que se obtenga sobre las células deformadas puede emplearse como apoyo al diagnóstico clínico, estableciendo un criterio sobre el nivel de afectación del organismo o la forma en que ha actuado algún medicamento empleado.

Entre las enfermedades que tienen esta característica se encuentra la Anemia Drepanocítica, que afecta alrededor del 3% de la población cubana, con mayor incidencia en la región oriental, y provoca la deformación de los eritrocitos existentes en la sangre debido al endurecimiento o polimerización de la hemoglobina que contienen. En Santiago de Cuba el estudio de las muestras de sangre de pacientes sicklémicos por parte del especialista para emitir un criterio sobre la cantidad de eritrocitos deformados presentes en las mismas se realiza con un microscopio. Este proceso es muy engorroso y propenso a errores, debido a las condiciones en que se desarrolla y requiere un alto costo en cuanto a tiempo y personal capacitado para realizarlo.

El objetivo de este trabajo es lograr un conteo automatizado de los eritrocitos deformados que puedan observarse en imágenes de muestras de sangre, empleando técnicas de procesamiento de imágenes digitales para detectar automáticamente los objetos de interés teniendo en cuenta su textura, aplicar métodos de evolución de contornos que permitan obtener los bordes de las células de forma eficiente y posteriormente realizar un análisis morfológico sobre los bordes detectados y determinar cuántas células están deformadas.

Los estudios se realizaron sobre 45 imágenes de muestras de sangre de pacientes sicklémicos preparadas en el Hospital General “Dr. Juan Bruno Zayas Alfonso”. En estas muestras se obtuvo la cantidad de eritrocitos deformados con una efectividad de un 75%, lo que se considera favorable dadas las condiciones de desarrollo del proceso actual. Los resultados fueron validados con el criterio de una especialista en 1er grado en MGI y Laboratorio Clínico del Departamento de Hematología Especial del hospital.

ABSTRACT.

Red blood cell deformation is the first exponent of the existence of certain sufferings or disease. The knowledge on the deformed cells can be used to help to the clinical diagnosis, because it allows to establish an opinion about the level of affectation of the organism or the answer to some used medication.

Among the disease that have this characteristic it is the Sickle Cell Anemia, that affects around the Cuban population's 3%, with more incidence in the oriental region, and it causes the deformation of the erythrocytes in the blood due to the hardening or polymerization of the hemoglobin that contain. In Santiago de Cuba the study of blood samples of sickle cell anemia patient for a specialist to emit an opinion on the quantity of deformed erythrocytes present in the same ones is carried out with a microscope. This process is very annoying and prone to errors, due to the conditions of development of the same one and it requires a high cost of time and qualified personnel to carry out it.

The goal of this work is to achieve an automated count of the deformed erythrocytes that can be observed in blood samples images, using digital image processing techniques to detect the objects of interest automatically considering their texture, to apply methods of contours evolution to obtain the borders of the cells in an efficient way and later on to carry out a morphological analysis of the detected borders and to determine how many cells are deformed.

The studies were carried out on 45 images of blood samples of sickle cell anemia patient prepared in the General Hospital "Dr. Juan Bruno Zayas Alfonso". In these samples the quantity of deformed erythrocytes was obtained with an effectiveness of 75%, this is considered favorable given the conditions of development of the current process. The results were validated with the approach of a specialist in 1er grade in MGI and Clinical Laboratory of the laboratory of Special Hematology of the hospital.

ÍNDICE.

INTRODUCCIÓN.	1
CAPÍTULO 1. ANTECEDENTES TEÓRICOS.	4
1.1 Introducción.	4
1.2 Detección de objetos en la imagen empleando texturas.	5
1.2.1 Medidas de textura empleadas.	7
1.2.2 Clasificación o agrupamiento de texturas.	14
1.2.3 Creación, unión y reducción de matrices de aprendizaje.	16
1.2.4 Clasificación a partir del conocimiento adquirido.	18
1.3 Evolución de contornos empleando Level Set.	19
1.3.1 Teoría de la evolución de interfaces.	20
1.3.2 Level Set estacionarios. Formulación de valores de frontera.	21
1.3.3 Level Set dependientes del tiempo. Formulación de valores iniciales.	23
1.3.4 Segmentación usando los Level Set.	35
1.4 Análisis de forma.	36
1.5 Conclusiones parciales.	37
CAPÍTULO 2. DETALLES DE IMPLEMENTACIÓN.	38
2.1 Introducción.	38
2.2 Extracción de características.	39
2.2.1 Momento de orden k.	41
2.2.2 Matriz de co-ocurrencia.	42
2.2.3 Modelos de Gabor.	43
2.3 Agrupamiento de texturas.	44
2.4 Creación de la matriz de aprendizaje.	47
2.5 Detección inicial de objetos.	49
2.6 Segmentación a partir de los métodos Level Set.	50
2.6.1 Algoritmo Level Set. Acercamiento matricial completo.	51
2.6.2 Algoritmo Fast Marching.	52

2.7 Análisis Morfológico.	54
2.8 Diseño del prototipo del sistema.	55
2.8.1 Interfaz del proceso de entrenamiento.	56
2.8.2 Interfaz del proceso de análisis.	58
CAPÍTULO 3. RESULTADOS Y VALIDACIÓN.	60
3.1 Introducción.	60
3.2 Condiciones iniciales para el estudio.	60
3.3 Selección del método de extracción de características.	61
3.4 Valoración de las estrategias de agrupamiento de características.	62
3.4.1 Estrategia de agrupamiento por K-means.	62
3.4.2 Estrategia de agrupamiento por β_0 -conexo.	65
3.5 Detección inicial de objetos de interés.	67
3.6 Resultados de la segmentación de los objetos en la imagen.	69
3.7 Calibración de los coeficientes morfológicos para la clasificación de formas en los eritrocitos de las muestras de sangre.	70
3.8 Conteo de eritrocitos deformados. Validación de resultados.	72
3.9 Conclusiones parciales.	76
CONCLUSIONES.	77
RECOMENDACIONES.	78
REFERENCIAS BIBLIOGRÁFICAS.	79
OTRAS BIBLIOGRAFIAS.	81

INTRODUCCIÓN.

La deformación de las células en la sangre es el primer exponente de la existencia de ciertos padecimientos o enfermedades. El conocimiento que se obtenga sobre las células deformadas puede emplearse como apoyo al diagnóstico de estas enfermedades, y sirve para establecer un criterio sobre el nivel de afectación que tiene el individuo o incluso sobre la forma en que ha actuado algún medicamento empleado en tratamientos específicos.

Entre las enfermedades que provocan deformaciones en las células que se encuentran en la sangre está la Anemia Drepanocítica (AD) o Sicklemia, una de las más frecuentes en el mundo, que presenta clínicamente eventos de obstrucciones vasculares, con el consiguiente daño de los órganos y los tejidos, se caracteriza por susceptibilidad a las infecciones incrementada, retraso del crecimiento, desarrollo de crisis de dolor por obstrucción vascular que pueden generar infartos de la médula ósea, huesos, y otros trastornos del organismo. En Cuba el 3.1% de la población es portadora de la hemoglobina anormal y en la región oriental llega a un 5%, por lo que es considerada una enfermedad común y la única patología hematológica de carácter genético que constituye todavía un problema social y de salud pública. Su prevalencia varía en cada uno de los subgrupos de población, de acuerdo con el antecedente genético, las condiciones ambientales y climáticas, el nivel de salud pública y social y los patrones matrimoniales.

Esta enfermedad se caracteriza por la deformación del glóbulo rojo, debido a que al desoxigenarse la hemoglobina que contiene, se forman agregados moleculares o polímeros que endurecen y distorsionan al mismo (sickling o deformación celular). Con la flexibilidad disminuida los glóbulos rojos pueden obstruir los pequeños capilares en la circulación, esto provoca la destrucción de los mismos más rápido que lo normal y ocasiona una oxigenación deficiente de los tejidos. Casi siempre las células afectadas adquieren forma de hoz o media luna y pueden ser observadas a través del microscopio.

En Santiago de Cuba el estudio de las deformaciones de los eritrocitos en pacientes sicolémicos se realiza por parte del especialista empleando muestras de sangre, que son observadas en un microscopio donde el mismo determina un estimado de la cantidad de células deformadas que tiene el paciente. Este proceso resulta ser muy engorroso y requiere un alto costo en cuanto a tiempo y personal capacitado para realizarlo, aun así el porcentaje de fiabilidad en los resultados es bajo debido a que el error humano en que incurre puede ser bastante alto teniendo en cuenta las condiciones de desarrollo del mismo.

Una forma de mejorar este proceso es empleando técnicas de procesamiento de imágenes digitales, garantizando la obtención de imágenes de las muestras de sangre con una cierta calidad y que contengan información representativa del proceso a valorar, a partir de las cuales se pueden aplicar métodos efectivos para realizar el conteo de las deformaciones antes mencionadas de forma automática. Esto permite que los especialistas puedan evaluar con más precisión y en menor tiempo la cantidad de eritrocitos deformados presentes en muestras de sangre de pacientes sicolémicos, y su esfuerzo se reduce a garantizar la calidad de las imágenes a valorar.

Por lo tanto se definió como objetivo general de esta investigación:

1. Determinar de forma automatizada la cantidad de eritrocitos deformados en una muestra de sangre.

Para esto se planteó la posibilidad de detectar automáticamente los objetos de interés (en este caso los eritrocitos) teniendo en cuenta su textura, y a partir de esta detección aplicar métodos de evolución de contornos que permitan obtener los bordes de las células de forma eficiente, para posteriormente realizar un análisis morfológico sobre los bordes detectados y determinar cuántos eritrocitos están deformados.

Para lograr este objetivo se definieron los siguientes objetivos específicos:

2. Determinar los métodos de extracción de características a aplicar.
3. Obtener la matriz de aprendizaje adecuada.
4. Determinar los métodos de clasificación a partir de la matriz de aprendizaje obtenida.

5. Determinar los métodos de segmentación de contornos a aplicar.
6. Determinar medidas a emplear para obtener el nivel de deformación celular.
7. Obtener un prototipo de software que posibilite el empleo de los métodos anteriores en la obtención del número de células deformadas dentro de una muestra microscópica de sangre.
8. Validar estadísticamente los resultados obtenidos.

El informe consta de resumen, introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y otras bibliografías empleadas.

En el Capítulo 1 se describen los métodos empleados para la extracción de características, la clasificación para detección y agrupamiento de texturas, unión y reducción de matrices, la segmentación empleando evolución de contornos y el análisis morfológico realizado sobre los contornos obtenidos.

En el Capítulo 2 se detallan cuestiones sobre la implementación de los algoritmos propuestos en el primer capítulo, se presenta el prototipo de software implementado que permite aplicar los mismos y se describe la metodología empleada para realizar la detección, segmentación y análisis de los objetos de interés

En el Capítulo 3 se presentan los resultados obtenidos con los métodos propuestos y se realiza una validación estadística que permite valorar la eficiencia del proceso.

Finalmente, en las conclusiones se resumen los resultados más importantes obtenidos como parte del estudio realizado y se dan las recomendaciones para perfeccionarlo en el futuro.

CAPÍTULO 1. ANTECEDENTES TEÓRICOS.

1.1 Introducción.

La deformación de las células en la sangre es el primer exponente de la existencia de ciertos padecimientos o enfermedades. El conocimiento que se obtenga sobre las células deformadas puede emplearse como apoyo al diagnóstico de estas enfermedades, y sirve para establecer un criterio sobre el nivel de afectación que tiene el individuo o incluso sobre la forma en que ha actuado algún medicamento empleado en tratamientos específicos. Entre las enfermedades que tienen esta característica se encuentra la Anemia Drepanocítica o Sicklemia, que provoca que los eritrocitos o glóbulos rojos que se encuentran en la sangre se deformen debido al endurecimiento o polimerización de la hemoglobina que contienen, con lo cual estas células sufren cambios en su forma que afectan el normal funcionamiento del organismo.

El objetivo de este trabajo es lograr un conteo automatizado de los eritrocitos deformados que puedan observarse en imágenes de muestras microscópicas de sangre de personas que padecen de Anemia Drepanocítica, un proceso que resulta engoroso y difícil de realizar de forma manual y que requiere de gran esfuerzo del especialista.

Para lograr el objetivo que persigue este trabajo se consideraron varias imágenes de muestras de sangre, donde pueden observarse ciertos componentes (eritrocitos, plaquetas, leucocitos y otros) con características visuales o texturas distintas. Se planteó la posibilidad de detectar automáticamente los objetos de interés, en este caso los eritrocitos, teniendo en cuenta su textura, y a partir de esta detección aplicar métodos de evolución de contornos que permitan obtener los bordes de las células de forma eficiente, para posteriormente realizar un análisis morfológico sobre los bordes detectados y determinar cuántas células están deformadas.

Como se pretende comprobar la factibilidad de realizar el proceso en esta forma se van a aplicar en un primer momento métodos de fácil comprensión y sencillos de programar, para en investigaciones posteriores valorar otros que sean más eficientes. En el caso de la detección empleando descriptores de textura se van a valorar tres métodos de

extracción de características: momento de orden k, matriz de co-ocurrencia y modelos de Gabor. Para el agrupamiento de los descriptores obtenidos se van a valorar los métodos de clasificación supervisada K-means y β_0 -conexo. Para la clasificación de los objetos a partir del conocimiento adquirido se emplearán los algoritmos k-NN y votación. Para la evolución de contornos se emplearán los algoritmos Fast Marching y Level Set y para el análisis morfológico se determinarán varias medidas, entre las que se encuentran los coeficientes elípticos y de compacidad.

En este capítulo se describen desde el punto de vista teórico todos los métodos y estrategias empleados en la investigación.

1.2 Detección de objetos en la imagen empleando texturas.

Para el procesamiento de las imágenes es necesario muchas veces determinar las regiones u objetos que la conforman, para poder evaluarlos y emitir un criterio. Para lograrlo existen diversos métodos, y la determinación de cuál emplear depende del problema en cuestión, es decir, de las características de los objetos en la imagen y las condiciones bajo las que fueron adquiridas las mismas. En la práctica no se usa un único método, más bien se trata de combinar varios para mejorar los resultados que se obtienen, ya sea en tiempo o precisión de los mismos.

Existen tres casos típicos según las imágenes a procesar [1]:

El primero es cuando los objetos y el fondo de las imágenes presentan intensidades de gris homogéneas, los valores para ambas clases son distinguibles y se pueden obtener fácilmente empleando métodos sencillos de procesamiento como la umbralización. Este tipo de imágenes no es común, se presentan en casos sintéticos o cuando el proceso de captura, el ambiente de iluminación y el proceso de formación de la imagen son controlados, lo que no es fácil de satisfacer en la práctica (ver Figura 1.1(a)).

El segundo caso es similar al anterior, las intensidades son homogéneas pero la iluminación no lo es y un mismo objeto presenta intensidades diferentes (ver Figura 1.1 (b)), por lo que no es posible recurrir solamente a esta información y se requieren otras fuentes, como por ejemplo los bordes existentes.

En el último caso las imágenes son más complejas y dentro de la misma se pueden definir texturas visuales que caracterizan a los objetos de interés o el fondo [1].



a) Homogeneidad uniforme b) Diferencia en la iluminación c) Presencia de texturas

Figura 1.1. Diferentes tipos de imágenes.

Las imágenes que se estudian en esta investigación son de muestras de sangre microscópicas y en ellas se pueden encontrar varios tipos de objetos diferentes en cuanto a sus texturas, como son las plaquetas, eritrocitos y artefactos provenientes del colorante empleado en la preparación de la muestra. El interés inicial se dirige solamente a los eritrocitos, por lo que para detectar los mismos en las imágenes se pensó en la utilización de métodos combinados, en este caso determinar las texturas correspondientes a estos objetos y a partir de las mismas utilizar métodos de segmentación más efectivos para detectar contornos y poder estudiarlos.

La textura es cualitativamente descrita por la percepción y la apariencia de una superficie (ver Figura 1.2), donde se pueden observar estructuras locales que se repiten y que según su período de repetición la textura en cuestión es fina o gruesa. Esto da una noción intuitiva de la inclinación de los valores de las medidas que se empleen, si son numéricamente pequeñas se corresponden con texturas finas y si son numéricamente altas con texturas gruesas.

Cualitativamente se puede definir la *textura* según Shapiro y Stockman [2] como "...tres ingredientes: 1.- algún 'orden' local es repetido sobre una región que es grande en contraste con el tamaño del orden, 2.- el orden consiste en la distribución poco aleatoria de partes elementales y 3.- las partes son entidades apenas uniformes que tienen aproximadamente las mismas dimensiones dentro de la región texturizada". Haralick [3] indica que: "Una textura está definida por la uniformidad, densidad, grosor, rugosidad,

regularidad, intensidad y direcciónabilidad de medidas discretas del tono y de sus relaciones espaciales”.

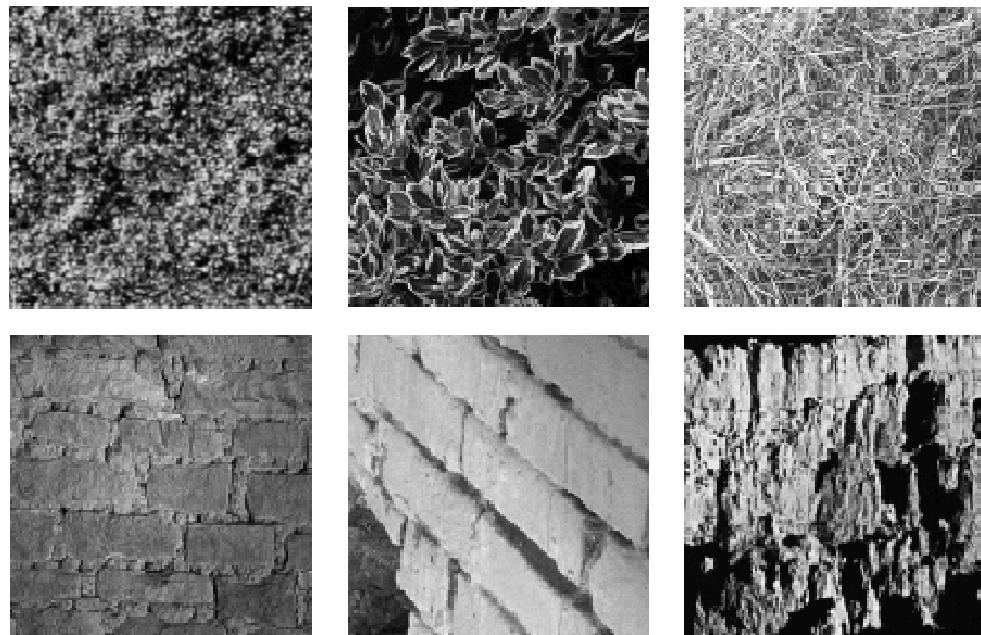


Figura 1.2. Ejemplo de texturas.

Computacionalmente la textura es una propiedad espacial perteneciente a una porción de la imagen determinada por la relación que existe entre un punto y sus vecinos, por lo que depende del tamaño de la vecindad de observación. Se puede entonces definir una textura T_{ij} perteneciente a la imagen I como un vector de medidas o rasgos característicos X_i calculado a partir del píxel p_{ij} en una vecindad de tamaño h , esto es:

$$T_{ij} = [X_1, X_2, \dots, X_n], \quad X_i = f_i(p_{ij}, h). \quad (1.1)$$

Donde f_i es la función que calcula la característica i -ésima en la ventana de tamaño $2*h+1$ con centro en p_{ij} .

1.2.1 Medidas de textura empleadas.

Los métodos de extracción de características a emplear para el trabajo pertenecen a dos enfoques clásicos para el estudio de texturas, el estadístico y elpectral, y no son los únicos, pero sí los más sencillos en cuanto a comprensión e implementación, por lo que se estableció su aplicación en la investigación teniendo en cuenta que la misma está dirigida en estos momentos a probar si es factible o no realizar el conteo

automatizado de eritrocitos deformados en muestras de sangre. En pruebas posteriores se pretende valorar algoritmos pertenecientes a estos enfoques que pueden ser más eficientes.

A continuación se detallan las medidas que forman parte del vector característico T_{ij} obtenidas empleando los tres métodos propuestos.

Momento de orden k.

La expresión del *momento k-ésimo* [4] en función de la media está dada por:

$$\mu_k = \sum_{i=0}^{L-1} (z_i - m)^k p(z_i) \quad (1.2)$$

Donde z_i es una variable aleatoria que indica la intensidad, $p(z_i)$ es el histograma de los niveles de intensidad en una región, L es el número de niveles posibles de intensidad, y

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (1.3)$$

representa la intensidad promedio o la media.

Mediante esta expresión se obtienen varias medidas de textura, entre las que están:

- **Media.** Es una medida de la intensidad promedio de la imagen, se calcula:

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (1.4)$$

- **Desviación Estándar.** Es una medida del contraste promedio de la imagen, se calcula:

$$\sigma = \sqrt{\mu_2(z)} = \sqrt{\sigma^2} \quad (1.5)$$

- **Descriptor de suavidad.** Es una medida del contraste del nivel de gris, describe la suavidad en la imagen, toma valores cercanos a 0 y 1 en áreas de intensidad constante o de gran variación respectivamente, se calcula:

$$R = 1 - \frac{1}{1 + \sigma^2} \quad (1.6)$$

- Tercer Momento. Mide la asimetría del histograma. Esta medida es 0 para histogramas simétricos bifurcados a la derecha (cerca del término medio) y negativa para histogramas bifurcados a la izquierda. Se calcula mediante la ecuación:

$$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i) \quad (1.7)$$

- Uniformidad. Mide la uniformidad de la imagen, es máxima cuando todos los niveles de grises son iguales y se calcula:

$$U = \sum_{i=0}^{L-1} p^2(z_i) \quad (1.8)$$

- Entropía. Mide la complejidad de la imagen y se calcula:

$$e = - \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \quad (1.9)$$

Matriz de co-ocurrencia.

La matriz de co-ocurrencia C de niveles de grises de una imagen I para una distancia d y en la orientación α , está definida según [1] como:

$$C_{d,\alpha}(i,j) = |\{(r,s), (t,v) : I(r,s) = i, I(t,v) = j\}| \quad (1.10)$$

donde $(r,s), (t,v) \in NxN$ y $|\cdot|$ es el cardinal del conjunto.

Cada elemento $C(i,j)$ de la matriz de co-ocurrencia es el número de veces que un píxel de valor i tiene una relación (d,α) con otro píxel de valor j , se cuenta cada píxel que se encuentra a una distancia d del píxel de referencia en la dirección del ángulo α y cada par de píxeles se cuenta dos veces, una vez a la derecha y otra vez a la izquierda para garantizar la simetría de la matriz, luego esta matriz se expresa como probabilidad y su expresión está dada por:

$$p_{d,\alpha}(i, j) = \frac{C_{d,\alpha}(i, j)}{\sum_{l,k=0}^{G-1} C_{d,\alpha}(l, k)} \quad i, j = 0, 1, \dots, G-1 \quad (1.11)$$

Donde $C_{d,\alpha}(i,j)$ es el valor de la matriz de co-ocurrencia en la posición (i,j) y G es el mayor nivel de gris de los píxeles en la imagen donde se obtuvo la matriz. El problema en este caso es cómo elegir el tamaño del desplazamiento d . Esto es considerado una gran desventaja en la aplicación de este método, ya que no hay una fórmula para determinar los valores óptimos del mismo, hasta el momento se toman de forma empírica [5].

Los rasgos de la textura que se obtienen a partir de la matriz de co-ocurrencia son:

- Contraste. Es una medida de la variación local de la imagen, se calcula:

$$CON = \sum_{i,j=0}^{G-1} |i - j|^2 p(i, j) \quad (1.12)$$

- Correlación. Mide la similitud entre píxeles vecinos, se calcula:

$$COR = \sum_{i,j=0}^{G-1} \frac{(i - \mu_i)(j - \mu_j)p(i, j)}{\sigma_i \sigma_j} \quad (1.13)$$

Donde el valor de la media está dado por:

$$\mu_i = \sum_{i,j=0}^{G-1} ip(i, j) \quad \mu_j = \sum_{i,j=0}^{G-1} jp(i, j) \quad (1.14)$$

y la varianza por:

$$\sigma_i^2 = \sum_{i,j=0}^{G-1} p(i, j)(i - \mu_i)^2 \quad \sigma_j^2 = \sum_{i,j=0}^{G-1} p(i, j)(j - \mu_j)^2, \quad \sigma_i^2 = \sigma_j^2 \quad (1.15)$$

- Uniformidad. Mide la homogeneidad de la imagen, es 1 para una imagen constante y se calcula:

$$ENG = \sum_{i,j=0}^{G-1} p(i, j)^2 \quad (1.16)$$

- Homogeneidad. Mide la probabilidad de que dos píxeles vecinos tengan el mismo nivel de gris y se calcula:

$$H = \sum_{i,j=0}^{G-1} \frac{p(i,j)}{1+|i-j|} \quad (1.17)$$

- Probabilidad Máxima. Mide la probabilidad máxima referida a $p(i,j)$, se calcula mediante la fórmula:

$$M = \max_{i,j=1 \dots G-1} (p(i,j)) \quad (1.18)$$

Entropía. Mide la complejidad de la imagen y se calcula:

$$E = -\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p(i,j) \log_2 p(i,j) \quad (1.19)$$

Modelos de Gabor.

Los modelos de Gabor emplean un banco de filtros de Gabor que brindan información de la imagen en cualquier orientación y frecuencia, cada conjunto de rasgos obtenido pertenece a un filtro distinto para destacar ciertas características de la imagen. El proceso de filtrado consiste en que se tiene una imagen de entrada $I(x,y)$, $x,y \in \Omega$ (Ω -toma como valores los puntos de la imagen) y es convolucionada con una función bidimensional de Gabor $g(x,y)$, $x,y \in \Omega$, obteniéndose la imagen de características $r(x,y)$, $x,y \in \Omega$ dada en [6] por:

$$r(x,y) = \iint_{\Omega} I(\xi,\eta) g(x-\xi, y-\eta) d\xi d\eta \quad (1.20)$$

Una función de Gabor bidimensional es una sinusoida Gaussiana [6]. La forma general de la función de Gabor está dada por la siguiente ecuación:

$$g_{\lambda,\theta,\psi}(x,y) = 10^{-\frac{x^2 + \gamma^2 + y^2}{2\sigma^2}} \cdot \cos(2\pi \frac{X}{\lambda} + \psi) \quad (1.21)$$

donde

$$\bullet \quad X = x \cdot \cos\theta + y \cdot \sin\theta \quad (1.22)$$

$$\bullet \quad Y = -x \cdot \sin\theta + y \cdot \cos\theta \quad (1.23)$$

▪ γ : Relación de aspecto horizontal/vertical o excentricidad del núcleo.

▪ θ : Ángulo de inclinación de ondas paralelas del filtro

- σ, λ, ψ : Parámetros relacionados con la amplitud, varianza de la onda y fase del filtro [6].

Un banco de filtros de Gabor se obtiene asignándoles valores a las variables antes mencionadas y el filtrado de la imagen está dado por la convolución de la misma con cada uno de ellos, se obtiene un resultado para cada uno. Entre las características que se obtienen, según [6], tenemos:

- Descriptor de suavidad. Es una medida del contraste del nivel de gris, describe la suavidad en la imagen, toma valores cercanos a 0 y 1 en áreas de intensidad constante o de gran variación respectivamente, se calcula:

$$R = 1 - \frac{1}{1 + \sigma^2} \quad (1.24)$$

- Desviación Estándar. Mide el contraste promedio de la imagen, se calcula:

$$\sigma = \sqrt{\sigma^2} \quad (1.25)$$

- Energía de activación. Es la energía emitida por el filtro y se calcula:

$$E = \sum_{i,j=0}^k G(i,j) \quad (1.26)$$

donde:

$$\sigma^2 = \sum_{i,j=0}^k G(i,j)(i - \mu)^2 \quad (1.27)$$

$$\mu = \sum_{i,j=0}^k iG(i,j) \quad (1.28)$$

Luego de obtener un vector de características de textura para cada píxel de una imagen se aplican algoritmos para saber en qué medida una textura es semejante a otra y lograr el agrupamiento de las que presenten características similares, con el objetivo de realizar el estudio posterior de los grupos obtenidos según el problema en cuestión, para esto los vectores característicos deben tener la misma dimensión. El vector característico T de medidas cuantitativas reales es un vector de \mathbb{R}^n en un espacio n -

dimensional y por tanto dos texturas serán semejantes en la medida en que sus vectores correspondientes se acerquen en el espacio, empleando alguna función de semejanza o similitud $\beta(T_i, T_j)$ entre texturas. En [7] se hace referencia a varios criterios o funciones de comparación válidos, de los cuales para este caso particular se empleó la distancia euclíadiana, que tiene la siguiente expresión:

Distancia Euclíadiana: [7]

$$\beta(T_i, T_j) = \sqrt{\sum_{k=1}^N (X_k(T_i) - X_k(T_j))^2} \quad (1.29)$$

donde $X_k(T_i)$ significa el rasgo k -ésimo de la textura i y N el número de rasgos.

Considerando que T_i es un conjunto de vectores $\{V_1, V_2, \dots, V_n\}$ tal que $V_k, k \in [1 \dots n]$ es un vector de $\Re^{[k]}$ compuesto por las medidas extraídas usando un método de extracción de características determinado. De esta manera dos texturas serán semejantes en la medida en que el promedio de las distancias para cada subconjunto V_k sea menor, por tanto:

$$\beta(T_i, T_j) = \frac{\sum_{k=1}^n Dist(V_k(T_i), V_k(T_j))}{n} \quad (1.30)$$

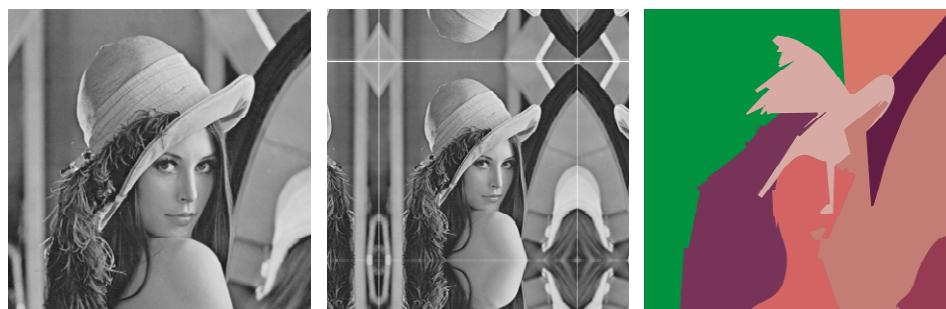
donde $V_k(T_i)$ significa el vector k -ésimo del conjunto de vectores T_i y la función $Dist$ es la propuesta en (1.29).

Empleando este criterio dos objetos son semejantes en la medida en que los valores que se obtienen con la función $\beta(T_i, T_j)$ son más pequeños, esto se conoce como criterio de diferencia entre objetos. En ocasiones es necesario la utilización de criterios de semejanza, donde la función $\beta(T_i, T_j)$ aporte valores mayores en la medida en que los objetos se parezcan. En este caso se define la siguiente función:

$$\beta(T_i, T_j) = \begin{cases} 1 & \text{si } \frac{\sum_{k=1}^n \text{Dist}(V_k(T_i), V_k(T_j))}{n} < \varepsilon \\ 0 & \text{en otro caso} \end{cases} \quad (1.31)$$

1.2.2 Clasificación o agrupamiento de texturas.

Para extraer las medidas de textura en una imagen se aplicará un algoritmo que realiza una exploración de todos los píxeles de la misma en un orden determinado, empleando una ventana de tamaño h que permita valorar un píxel y su vecindad. El vector de características hallado se le asigna al píxel central de la ventana. Este algoritmo presenta como límite la no definición de la ventana en los puntos que se encuentren a una distancia $h/2$ de los límites de la imagen, para lo que se añaden celdas artificiales a la imagen original, como se muestra en la Figura 1.3(b). Como resultado de este proceso se obtiene un arreglo de texturas $[T_1, T_1, \dots, T_{n \times m}]$ tal que $T_i = [X_1, X_2, \dots, X_N]$ representa el vector característico de la textura i -ésima y X_i es una de las medidas de texturas descritas anteriormente.



a) Imagen original. b) Celdas artificiales. c) Agrupamiento de la imagen (a).

Figura 1.3. Proceso de agrupamiento.

Cuando se cuenta con el conjunto de vectores característicos de textura de una imagen es de interés el agrupamiento de los mismos para determinar zonas de características similares (ver Figura 1.3(c)) y dividir la imagen en regiones no solapadas, teniendo en cuenta un criterio de igualdad o semejanza entre los vectores. En este caso se tiene un universo de objetos (los vectores) que se desea dividir en clases que no se conocen de antemano, lo que hace conveniente el uso de algoritmos de clasificación no supervisada que permitan obtener particiones o cubrimientos del universo en los que la semejanza intra-grupo sea máxima y la semejanza inter-grupo sea mínima. En este caso particular

se emplearon dos algoritmos de clasificación no supervisada, el K-means y el β_0 -conexo, ambos se detallan a continuación.

Clasificación por K-means.

Este es el algoritmo de clasificación no supervisada restringida que se utiliza en la mayoría de la bibliografía para agrupamiento de texturas, tiene como entrada el universo de objetos y k que representa el número de grupos a formar, la distancia de un objeto a un grupo es la distancia del objeto al centroide u objeto promedio de dicho grupo. Es sencillo de implementar y comprender, pero tiene como desventaja que hay que fijar la cantidad de grupos que se obtienen a priori para poder aplicarlo, de ahí que sea restringido y que depende de la inicialización de sus k primeros centroides, lo que trae consigo que para una misma entrada de datos se obtengan resultados diferentes.

Los grupos formados a partir de este método son esféricos, lo que restringe la forma de estos, pero garantiza que todos los objetos pertenecientes a un mismo grupo tengan su representante en el centroide, es decir, que todo objeto que sea semejante al grupo lo será al centroide, teniendo en cuenta alguna función de semejanza. Para más detalle sobre este algoritmo se puede consultar [7].

Clasificación por β_0 -conexo.

En este algoritmo no se fija a priori la cantidad de grupos a obtener, por lo que es un algoritmo de clasificación no supervisada libre, sólo depende de un umbral (β_0) que indica en qué medida se van a parecer dos objetos. Tiene como entrada el universo de objetos y el umbral β_0 . Emplea una matriz de semejanza donde en cada posición $S(i,j)$ se tiene la semejanza del objeto i al objeto j . Partiendo de esta matriz se construye un grafo de β_0 -semejanza que como vértices tiene los objetos y como aristas la semejanza entre ellos. El valor de β_0 en el trabajo se definió de forma experimental en casi todas las ocasiones en que se empleó el algoritmo, aunque en algunos casos se utilizó la expresión siguiente, descrito en [7]:

$$\beta_0 = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N \beta(O_i, O_j). \quad (1.32)$$

Para este método de agrupamiento el cálculo del representante de cada grupo no es trivial y una de las limitaciones radica en el fenómeno de encadenamiento [7], que provoca que a un mismo grupo puedan pertenecer objetos diferentes. Para más detalles sobre el algoritmo puede consultarse [7].

El problema de detección de objetos en la imagen puede verse como un problema de detección de texturas siempre y cuando los objetos a detectar se caractericen por un conjunto de texturas. Si se conocen las texturas presentes en los objetos de estudio de la imagen se está en presencia de un problema de clasificación supervisada, en el que se tiene un universo de objetos dividido en clases (las texturas pertenecientes al objeto de estudio en la imagen y las que no pertenecen), y se tienen muestras de cada una de ellas. Para resolver esto se necesita crear una base de conocimiento o matriz de aprendizaje y posteriormente clasificar a partir del conocimiento adquirido.

1.2.3 Creación, unión y reducción de matrices de aprendizaje.

En muchos problemas surge la necesidad práctica de contar con “buenas” muestras, las cuales forman una matriz llamada matriz de aprendizaje (*MA*), donde las filas representan los objetos (las texturas) y las columnas los rasgos (cada componente del vector característico *T*) seguido de un c-uplo de pertenencia que no es más que un vector booleano donde el elemento $c(i,j)=1$ indica que el objeto O_i pertenece a la clase j . La cantidad de objetos en la muestra es un factor importante ya que denota un supuesto mayor conocimiento acerca del fenómeno que se estudia, aunque no siempre una mayor cantidad de objetos en la muestra la mejora, más bien debe buscarse variedad pues es más importante conocer todas las variantes en que se manifiestan los objetos que tener muchos ejemplos de una de esas variantes.

En esta investigación se decidió realizar un agrupamiento previo de las texturas presentes en la imagen aplicando los algoritmos descritos anteriormente. Los representantes de estos grupos forman inicialmente la matriz de aprendizaje y el usuario define de forma interactiva las clases de grupos de texturas representativos de

objetos de interés. Como resultado de este proceso se obtiene una matriz de aprendizaje suficientemente representativa de las texturas en la imagen.

Para obtener una buena matriz de aprendizaje se realiza un proceso de entrenamiento para varias imágenes, dado que es difícil que en una sola imagen se hallen todas las texturas que puedan valorarse como pertenecientes a objetos de interés. Cada matriz de aprendizaje obtenida durante el procesamiento de cada imagen contiene información nueva, pero también contiene información que ya existía o no contiene alguna información adquirida con otras imágenes, por lo que se va realizando la unión y reducción de las matrices para garantizar la minimización de la redundancia y pérdida de la información. Este proceso no es trivial, pues se deben incorporar solamente muestras que representen texturas nuevas, lo que depende de las funciones de semejanza y del criterio que se utiliza para determinar qué muestra es lo suficientemente nueva como para incorporarla a la matriz resultado. A continuación se describe este proceso.

Unión y reducción de matrices de aprendizaje.

Los criterios utilizados para la unión de las matrices tienen que garantizar la obtención de la nueva información y el no crecimiento ilimitado de la misma en pos de que sea lo más representativa posible. El algoritmo que se emplea para unir dos matrices tiene como idea central ir insertando en la primera los objetos de la segunda, siempre y cuando se garantice que el objeto va a aportar nueva información a la clase a la que pertenece.

La unión de dos matrices en una tiene como resultado una matriz de mayor tamaño, y si el proceso es repetitivo pronto las matrices resultantes alcanzarán dimensiones no deseadas, por lo que se propone la reducción de las mismas después de cada unión a partir de un algoritmo propuesto en [7] y detallado en el Capítulo 2 epígrafe 2.4. Para la reducción de matrices de aprendizaje se aplica un algoritmo de agrupamiento a cada clase y para cada grupo obtenido se selecciona el objeto más representativo. Este proceso da como resultado un subconjunto de vectores para cada clase, los demás son eliminados. El algoritmo empleado para el agrupamiento por clase debe garantizar la mayor cohesión posible entre los objetos de un mismo grupo para evitar la pérdida de

información, por lo que en este trabajo se decidió el empleo del algoritmo β_0 -compacto que en comparación con el β_0 -conexo el efecto de encadenamiento es más limitado.

1.2.4 Clasificación a partir del conocimiento adquirido.

Luego de tener definida una matriz de aprendizaje se pueden determinar cuáles son las texturas que pertenecen al objeto y cuáles no, se está en presencia entonces de un problema de clasificación supervisada. Existen numerosos algoritmos que dan solución a este tipo de problema, más o menos eficientes en dependencia del contexto en que sean aplicados. Particularmente se emplearon dos de los clásicos propuestos en la literatura, el algoritmo k-NN o k vecinos más cercanos y el algoritmo de votación.

Algoritmo k-NN.

La idea esencial de este algoritmo es la distancia entre los objetos. Al ubicar los objetos en el espacio, se decide por una u otra clase en dependencia de cuán cercanos se encuentren los objetos respecto a ciertos objetos de los ya ubicados en las diferentes clases definidas en el problema. La selección de esos objetos "distinguidos" entre los ya clasificados se realiza de varias formas, todas basadas en la idea fundamental de este modelo: la cercanía entre las representaciones de objetos en el espacio seleccionado con las características antes mencionadas.

Tiene como entrada el objeto a clasificar (O) y la cantidad de vecinos más cercanos (k). Este algoritmo es fácil de implementar, ofrece una buena clasificación, es lineal con respecto a la cardinalidad del conjunto de entrenamiento, permite obtener clases solapadas y trabaja con pocos ejemplos de entrenamiento. Para más detalle ver [7].

Algoritmo de votación.

El algoritmo de votación sigue la lógica de la regla de decisión del pensamiento humano, en el sentido en que casi en ningún caso se tienen en cuenta todos los rasgos que describen un objeto o fenómeno dado, sino más bien hay una tendencia al análisis de subconjuntos de rasgos. En este algoritmo se emiten conclusiones parciales en forma de voto sobre la base de los mismos, que permiten posteriormente, con la consecución de otras conclusiones de este tipo para otros subconjuntos de rasgos, llegar a una conclusión final. Estos son conocidos como conjuntos de apoyo que podrán

ser definidos por el usuario o por algún método de selección de variables. Para este problema el sistema de conjunto de apoyo estará formado por las clases de medidas utilizadas en el cálculo de las características de textura y la regla de decisión que se empleará será mayoría holgada. Para más detalles ver [7].

Hasta este punto se tiene una matriz de aprendizaje lo suficientemente representativa como para detectar regiones pertenecientes a objetos de interés en la imagen, que servirán como punto de partida a los métodos de segmentación basados en la evolución de contornos, que detectarán los bordes de manera tal que se pueda proceder a un análisis de formas.

A continuación se detalla primeramente todo lo relacionado con los métodos de segmentación basados en la evolución de contornos, y posteriormente el análisis de forma para reconocer un tipo de objeto sobre otros posibles.

1.3 Evolución de contornos empleando Level Set.

Los Level Set (Conjunto de Niveles) son métodos de segmentación basados en la evolución de contornos, son modelos dinámicos que evolucionan su forma en espacios n-dimensionales bajo la influencia de fuerzas externas e internas, minimizando una función de energía y fueron propuestos con efectividad por primera vez por Kass y Terzopoulos [8]. El concepto de contorno activo se genera a partir de la idea de adaptar dinámicamente contornos o superficies siguiendo propiedades físicas como la elasticidad, suavidad, dureza y otras. La evolución espacial del contorno activo de una curva inicializada depende de la imagen y/o de la parametrización de la misma. La dinámica de éste se detiene cuando el mismo alcanza los límites del objeto.

Estos modelos están basados en la teoría propuesta por Osher y Sethian en [9] de evolución de la curva por su curvatura media y por una fuerza externa calculada a partir de la imagen. Representan una poderosa herramienta para estudiar el movimiento de interfaces conociendo un modelo de la velocidad normal de esta.

1.3.1 Teoría de la evolución de interfaces.

Tomando la separación de una región de otra, ya sea en una curva en dos dimensiones o en una superficie que se mueve en el sentido de su dirección normal (orientada en sentido normal con respecto al exterior o al interior de la curva) mediante una función de velocidad F conocida, el objetivo es seguir el movimiento de esta interfaz, es decir, su evolución. Solamente interesa el movimiento de la interfaz en su dirección normal, por eso ignoraremos el movimiento tangencial de esta interfaz.

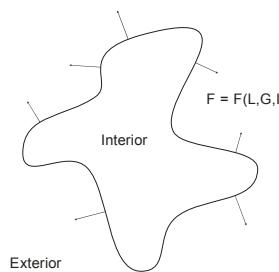


Figura 1.4. Curva propagándose con una velocidad F en dirección de la normal.

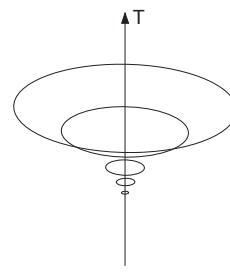


Figura 1.5. Propagación del frente en el espacio.

La función de velocidad F depende de tres tipos de argumentos.

$$F = F(L, G, I) \quad (1.33)$$

donde:

- L : Propiedades locales del frente, que son determinadas a través de informaciones geométricas locales, como son la curvatura y la dirección normal.
- G : Propiedades globales del frente, que dependen de la forma y posición del mismo. Por ejemplo la velocidad puede depender de integrales sobre el frente y/o ecuaciones diferenciales asociadas.
- I : Propiedades independientes, son aquellas que no dependen de la forma del frente.

Dada la velocidad F y la posición inicial de una interfaz, el objetivo es rastrear su posición en cada momento. La primera tarea será formular este problema de evolución desde una perspectiva *Euleriana* en la cual el sistema de coordenadas subyacente permanezca fijo (ver Figura 1.5).

1.3.2 Level Set estacionarios. Formulación de valores de frontera.

Dada una curva $\gamma(t=0)$ que se expande con una velocidad F en su dirección normal, a partir de su posición inicial cada cierto tiempo en el espacio, el objetivo es poder determinar una función que caracterice el movimiento de la curva en cualquier instante de tiempo t y de esta forma poder simular su movimiento para una función de velocidad determinada. El frente se considera como un conjunto de partículas que se mueven a partir de las condiciones dadas, como se muestra en la siguiente figura:

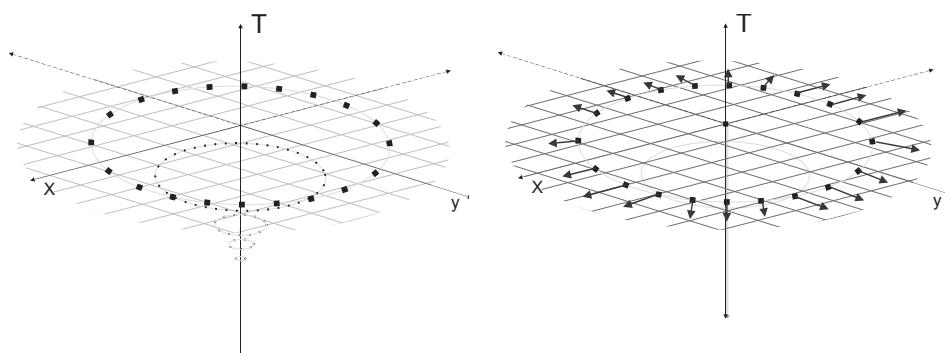


Figura 1.6. Movimiento de una partícula del frente en el espacio.

Figura 1.7. Cálculo del tiempo de cruce en (x,y) para la expansión de un frente con $F>0$.

En este caso la función de velocidad $F>0$, por tanto el frente siempre se mueve hacia “fuera”. La función de velocidad puede ser positiva o negativa, pero durante la evolución no puede cambiar de signo, sea uno u otro debe mantenerse. Debido a esto un método para caracterizar la posición de las partículas del frente en expansión es calcular el tiempo de arribo t que cada una se demora al pasar por un punto al expandirse, a partir de la función $T(x,y)$, como se muestra en la Figura 1.7.

Considerando que la *velocidad = distancia/tiempo* (ver Figura 1.8), la ecuación para la función de arribo $T(x,y)$ quedaría:

$$F = \frac{x}{T(x,y)} \quad (1.34)$$

Donde

- F es la velocidad.
- $T(x,y)$ es el tiempo.

- x es la distancia.

Despejando la distancia de la ecuación y derivando la expresión resultante respecto a la misma, quedaría la ecuación de la siguiente forma:

$$x = F^* T(x,y) \quad (1.35)$$

$$1 = F \frac{dT}{dx} \quad (1.36)$$

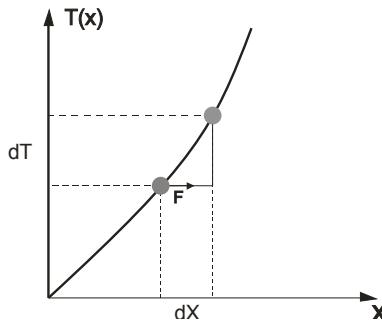


Figura 1.8. Configuración para la formulación de los valores de frontera.

En múltiples dimensiones, ∇T es ortogonal a los Level Set de T y de manera similar al caso unidimensional, esta magnitud es inversamente proporcional a la velocidad. Es decir,

$$|\nabla T|F = 1, \quad T = 0 \text{ en } \Gamma \quad (1.37)$$

donde Γ es la localización inicial de la interfaz.

Por tanto, el movimiento del frente es caracterizado como la solución a un problema de valores de contorno. Si la función de velocidad F depende solamente de la posición, entonces la ecuación será de tipo Eikonal [10]. Para ejemplificar lo anteriormente mencionado observe la siguiente figura donde se muestra la superficie de arriba $T(x,y)$ de un frente circular expandiéndose con velocidad constante $F=1$.

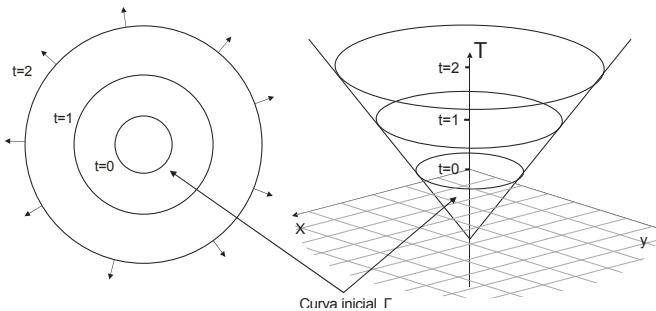


Figura 1.9. Transformación del movimiento de un frente a un problema de valores de contorno.

1.3.3 Level Set dependientes del tiempo. Formulación de valores iniciales.

Si por el contrario, el frente ahora se mueve con una velocidad F que no es estrictamente positiva ni negativa, puede ir hacia adelante o hacia atrás indistintamente, y por lo tanto puede pasar por encima de un punto (x,y) varias veces. Así, el tiempo del cruce $T(x,y)$ no es una función de valor único. Considerando una malla de dimensiones $N \times N$ colocada en un espacio tridimensional, si está tendida sobre el plano XY con centro en $(0,0)$ y es levantada por su punto medio en el sentido de la normal a este plano quedaría como se muestra en la Figura 1.10:

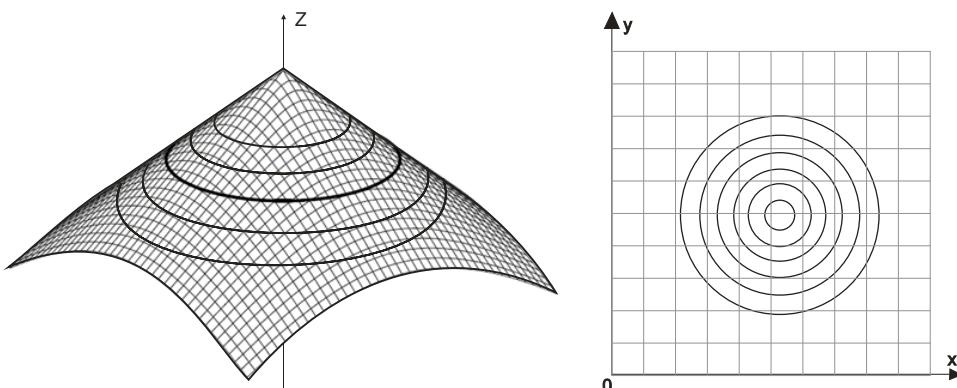


Figura 1.10. Conjunto de niveles en un espacio tridimensional.

Figura 1.11. Proyección del Level set en el plano XY .

Como se puede observar en esta figura, donde quiera que el plano XY corte a la malla se genera un conjunto de puntos que describen una curva en un mismo nivel, es decir, para un mismo valor de Z . Si se agrupan por niveles entonces se formaría una familia de curvas que estaría estrechamente relacionada con la malla que las genera. Una mirada desde la coordenada Z hacia abajo permite ver la proyección de éstas en el plano XY como se muestra en la Figura 1.11.

Dado que la malla queda definida por esta familia de curvas, el movimiento de las mismas en la dirección normal al plano XY con una velocidad F , generaría su movimiento en esa misma dirección. El movimiento de las partículas está caracterizado para un instante de tiempo t por la función $z = \phi(\vec{v}, t)$ donde $\vec{v} = (x, y)$ son las coordenadas de cada una de éstas. Luego se puede asociar la evolución de esta función ϕ a la propagación del frente a través de un problema de valor inicial dependiente del tiempo. En cualquier instante de tiempo t , el frente estará dado por el Level Set cero de la función Level Set ϕ (el Level Set cero es el nivel de cruce por cero) vea la Figura 1.12. Esta representación se basa en una formulación Lagrangiana donde los puntos son colocados directamente en la interfaz y se mueven con ella.

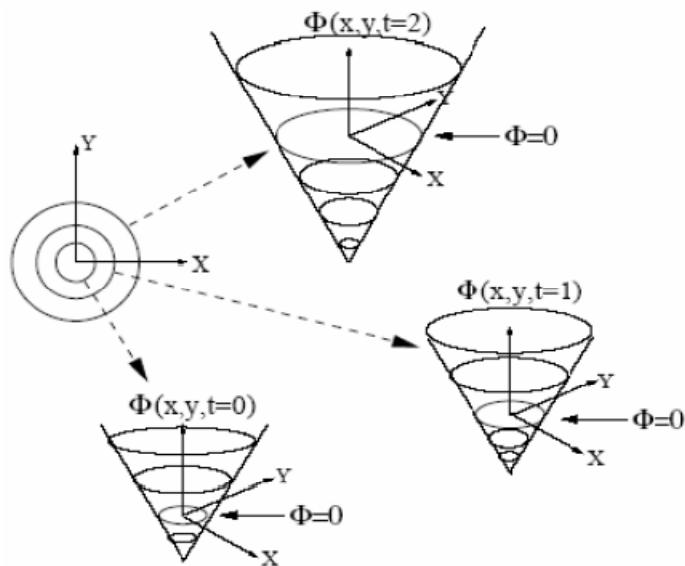


Figura 1.12. Transformación del movimiento del frente en un problema de valores iniciales.

Para deducir una ecuación del movimiento para esta función Level Set donde el Level Set cero de ϕ describa la evolución del frente, se necesita que el valor de las partículas pertenecientes a éste que describen un camino $x(t)$ sean siempre cero, y por tanto:

$$\phi(x(t), t) = 0 \quad (1.38)$$

Si se deriva respecto a t aplicando la regla de la cadena la expresión resultante quedaría:

$$\phi_t + F|\nabla\phi| = 0 \quad (1.39)$$

dado $\phi(x,t=0)$ como condición inicial.

Se pueden resumir estas dos perspectivas de la siguiente manera. Sea Γ una curva que se propaga en el plano en el sentido de su dirección normal con velocidad F tal que $\Gamma(t)$ es la posición del frente en el instante t . Luego se debe solucionar:

Formulación del valor de contorno/frontera:

$$|\nabla T|F = 1$$

$$\text{Frente} = \Gamma(t) = \{(x,y) | T(x,y)=t\} \quad (1.40)$$

para $F>0$.

Formulación de valor inicial:

$$\phi_t + F|\nabla\phi| = 0$$

$$\text{Frente} = \Gamma(t) = \{(x,y) | \phi(x,y,t)=0\} \quad (1.41)$$

para F arbitraria.

Formulación fundamental en la teoría de curvas.

Siendo γ una curva inicial cerrada y suave en \mathbb{R}^2 y $\gamma(t)$ el único parámetro de la familia de curvas generadas por el movimiento de $\gamma = \gamma(t=0)$ a lo largo del campo de su vector normal con velocidad F , donde F es una función escalar dada. De esta forma se tiene que $F = \vec{n}\vec{x}_t$, donde \vec{x}_t es el vector de posición de la curva, t es el tiempo y \vec{n} es la normal unitaria de la misma..

Considerando una descripción parametrizada de este movimiento y el caso en que la función F depende solamente de una curvatura local k de la curva, es decir, $F=F(k)$, el vector de posición $\vec{x}(s,t)$ parametriza a γ en el tiempo t . La parametrización de la normal hacia fuera y de la curvatura está dada por $\vec{n}(s,t)$ y $k(s,t)$ respectivamente. El rango de variación de s es $0 \leq s \leq S$, y las condiciones de límite periódico están dadas por

$\vec{x}(0, t) = \vec{x}(S, t)$. La curva es parametrizada para que el interior esté a la izquierda en la dirección del crecimiento de s (Figura 1.13).

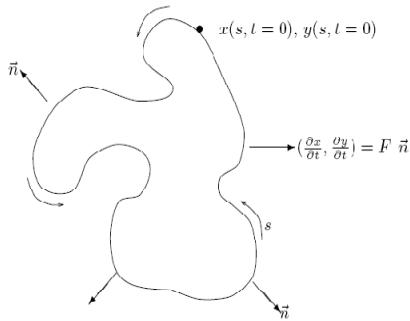


Figura 1.13. Vista parámetrica de la curva.

Las ecuaciones de movimiento pueden ser escritas en términos de los componentes individuales de $\vec{x}(x, y)$ como:

$$\begin{aligned} x_t &= F \left[\frac{y_{ss}x_s - x_{ss}y_s}{(x_s^2 + y_s^2)^{3/2}} \right] \left(\frac{y_s}{(x_s^2 + y_s^2)^{1/2}} \right), \\ y_t &= -F \left[\frac{y_{ss}x_s - x_{ss}y_s}{(x_s^2 + y_s^2)^{3/2}} \right] \left(\frac{x_s}{(x_s^2 + y_s^2)^{1/2}} \right) \end{aligned} \quad (1.42)$$

donde la expresión parametrizada de la curvatura dentro de la función de velocidad es:

$$k = \frac{y_{ss}x_s - x_{ss}y_s}{(x_s^2 + y_s^2)^{3/2}} \quad (1.43)$$

la expresión parametrizada de la normal está dada por:

$$\vec{n} = \frac{\vec{y}_s - \vec{x}_s}{(\vec{x}_s^2 + \vec{y}_s^2)^{1/2}} \quad (1.44)$$

y x_t denota la derivada de x con respecto a t .

El movimiento del frente a partir de esta formulación provoca que en ciertos puntos se desarrolle esquinas donde la normal no está definida.

El frente inicial dado por la función $f(x)$ con f y f' periódicas en $[0,1]$ tiene una propagación como la mostrada en la gráfica para todo tiempo t . Sea ψ la altura de la propagación de la función en un tiempo t , y así $\psi(x,0) = f(x)$. La tangente de (x,ψ) es $(1,\psi_x)$. La Figura 1.14 referencia el cambio en la altura V en una unidad de tiempo en relación con la velocidad F en dirección de la normal:

$$\frac{V}{F} = \frac{(1+\psi_x^2)^{1/2}}{1} \quad (1.45)$$

y así la ecuación de movimiento queda:

$$\psi_t = F(1+\psi_x^2)^{1/2} \quad (1.46)$$

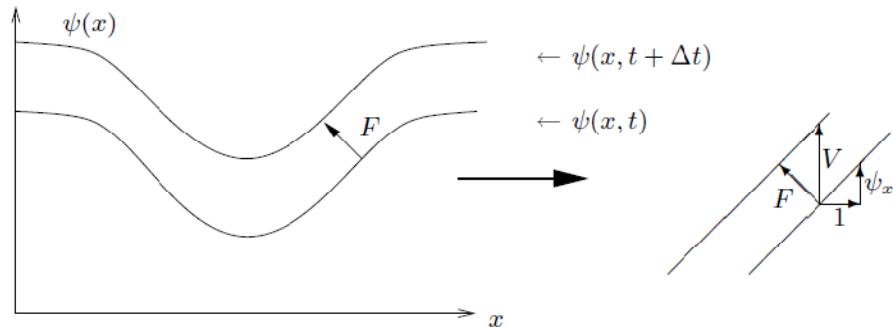


Figura 1.14. Variables de propagación gráficas.

Usando la función de velocidad $F(k) = 1 - k\varepsilon$ y la fórmula $k = \frac{-\psi_{xx}}{(1+\psi_x^2)^{3/2}}$ tenemos:

$$\psi_t = \left(1 + \varepsilon \frac{\psi_{xx}}{1 + \psi_x^2}\right) \cdot (1 + \psi_x^2)^{1/2},$$

$$\psi_t - (1 + \psi_x^2)^{1/2} = \varepsilon \frac{\psi_{xx}(1 + \psi_x^2)^{1/2}}{(1 + \psi_x^2)^{3/2}},$$

$$\psi_t - (1 + \psi_x^2)^{1/2} = \varepsilon \frac{\psi_{xx}}{1 + \psi_x^2}. \quad (1.47)$$

$$u_t + \left[- (1 + u^2)^{1/2} \right]_x = \varepsilon \left[\frac{u_x}{1 + u^2} \right]_x. \quad (1.48)$$

El término a la derecha es conocido como término de viscosidad para la función de movimiento, esta ecuación es considerada una “ley de conservación hiperbólica” [9] dado que tiene la siguiente forma:

$$u_t + [G(u)]_x = \varepsilon u_{xx}, \quad (1.49)$$

donde $G(u) = -(1+u^2)^{1/2}$, La interpretación física de este término describe el movimiento de un material u con una función de flujo $G(u)$, lo que hace posible el uso de herramientas numéricas para la aproximación del gradiente de este campo que permiten desarrollar esquemas como los dados en las ecuaciones (1.40) y (1.41) de valor inicial y de valor de contorno respectivamente.

Otra vista en pos de la solución de nuestras ecuaciones de movimiento (1.40) y (1.41) recae sobre la solución de viscosidad de las ecuaciones de Hamilton dadas en [9], basada en el comportamiento del movimiento, que resulta ser una forma más apropiada de caracterizar la solución débil correcta. Si la función de velocidad F en las ecuaciones (1.40) y (1.41) sólo depende de la posición x, y y de las primeras derivadas de Φ , estas ecuaciones pueden considerarse como casos particulares de la ecuación más general de Hamilton–Jacobi:

$$\alpha u_t + H(Du, x) = 0 \quad (1.50)$$

donde $H(Du, x) = F|\Delta u| - (1-\alpha)$, y α es cero o uno. Du representa la derivada parcial de u en cualquier variable por ejemplo, u_x y u_y . Si se añade un término de viscosidad, se tendría:

$$\alpha u_t + H(Du, x) = \varepsilon \Delta u, \quad (1.51)$$

Donde ε es una constante positiva y es posible emplear toda la teoría demostrada para las ecuaciones de Hamilton dadas en [9].

De las dos perspectivas antes expuestas, ley de conservación hiperbólica y las ecuaciones de Hamilton–Jacobi, Sethian en [9], propone el siguiente esquema numérico para la solución de la ecuación de Hamilton (1.50):

$$\alpha u_i^{n+1} = u_i^n - \Delta t g\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}, \frac{u_{i-1}^n - u_i^n}{\Delta x}\right) \quad (1.52)$$

donde g es una de las funciones numéricas de flujo deducida a partir de la ley de conservación hiperbólica dada por:

$$g(u_1, u_2) = (\max(u_1, 0)^2 + \min(u_2, 0)^2)^{1/2}, \quad (1.53)$$

y los términos $\frac{u_i^n - u_{i-1}^n}{\Delta x}, \frac{u_{i-1}^n - u_i^n}{\Delta x}$ son esquemas regresivos y progresivos para la

resolución del diferencial. Estos son deducidos a partir de la discretización del espacio en una red o malla donde $\Delta x = h$ y $\Delta t = k$. Así cada punto en la red puede ser representado por el par ordenado (i, n) correspondiente al punto (ih, nk) (ver Figura 1.15).

La ecuación $u_t + u_x = 0$ tiene entonces varias formas de aproximarse, se reescribe $u_t = -u_x$. Para la parte izquierda de la ecuación, la solución u en el tiempo $t + \Delta t$ puede ser desarrollada como una serie de Taylor en el tiempo alrededor del punto (x, t) :

$$u(x, t + k) = u(x, t) + u_t(x, t)k + O(k^2), \quad (1.54)$$

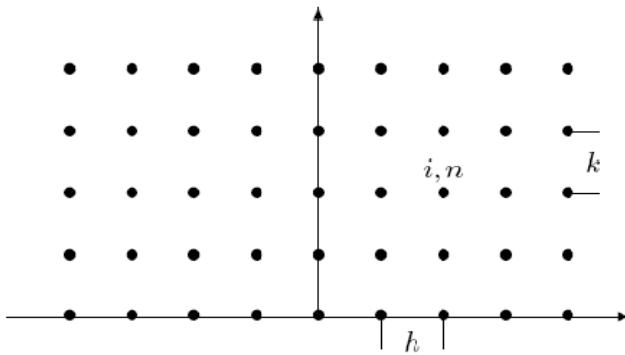


Figura 1.15. Discretización del dominio.

Donde la expresión $O(k^2)$ incluye todos los términos de orden k^2 o mayores. Reacomodando esta expresión, la derivada de u con respecto al tiempo en el punto (x, t) queda de la siguiente manera:

$$u_t = \frac{u(x, t + k) - u(x, t)}{k} + O(k^2). \quad (1.55)$$

Esto es conocido como operador de diferencias progresivas para la variable de tiempo t , porque se ha usado una serie de Taylor que adelanta en el tiempo la aproximación u_t .

Se define la notación:

$$D^{+t}u \equiv \frac{u(x, t+k) - u(x, t)}{k}, \quad (1.56)$$

Luego reescribimos la ecuación (1.55) como:

$$u_t = D^{+t}u + O(k) \quad (1.57)$$

En el caso de la derivada parcial u_x se definen los operadores:

$$\begin{aligned} D^{+x}u &\equiv \frac{u(x+h, t) - u(x, t)}{h} \\ D^{-x}u &\equiv \frac{u(x, t) - u(x-h, t)}{h} \\ D^{0x}u &\equiv \frac{u(x+h, t) - u(x-h, t)}{2h} \end{aligned} \quad (1.58)$$

De la misma manera se pueden construir desarrollos de la serie de Taylor de diferencias progresivas, regresivas, o central en x para el valor u alrededor del punto (x, t) para producir:

$$u_x = D^{+x}u + O(k), u_x = D^{-x}u + O(k), u_x = D^{0x}u + O(k^2) \quad (1.59)$$

La última es una aproximación más precisa. Estas conducen a tres esquemas distintos para computar la solución de la ecuación $u_t = -u_x$. Para u_i^n solución calculada en el tiempo nk y el punto ih se obtienen los esquemas:

- i. Esquema Progresivo: $u_i^{n+1} = u_i^n - kD^{+x}u_i^n,$
- ii. Esquema Regresivo: $u_i^{n+1} = u_i^n - kD^{-x}u_i^n,$
- iii. Esquema Central: $u_i^{n+1} = u_i^n - kD^{0x}u_i^n.$

En este punto para el uso de la función de flujo g mostrada en (1.60) y obtenida a partir de (1.52) y (1.53) debe seleccionarse el esquema apropiado a utilizar y depende del sentido en que se propaga la información. Considerando los tres operadores de diferencia siguientes:

- D^{+x} calcula el nuevo valor i usando la información de i e $i+1$; así la información para la solución se propaga de derecha a izquierda.
- D^{-x} calcula el nuevo valor i usando la información de i e $i-1$; así la información para la solución se propaga de izquierda a derecha.
- D^{0x} calcula el nuevo valor i usando la información de i , $i+1$ e $i-1$; así la información para la solución se propaga de ambos lados.

La función de flujo g selecciona el operador de diferencia que usará entre los tres definidos, en dependencia de los valores que recibe como argumentos, que ya han sido calculados y que se encuentran en sentido contrario a la propagación del frente.

$$g\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}, \frac{u_i^n - u_{i-1}^n}{\Delta x}\right)^2 = \max\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}, 0\right)^2 + \min\left(\frac{u_{i-1}^n - u_i^n}{\Delta x}, 0\right)^2 \quad (1.60)$$

A partir de todo lo anterior se explican las formulaciones de cada caso en particular, la formulación de valor de contorno $|\nabla T|F=1$ y la de valor inicial $\phi_t + F|\nabla\phi|=0$.

Problema de valor inicial: el método Level Set.

A continuación se ilustra el esquema más básico para el problema de valor inicial. Se comienza con el caso de una sola dimensión, y por lo tanto $H(u)=\sqrt{u^2}$. Otra vez simplemente se aplica el esquema dado por la ecuación (1.53) para la velocidad $F=1$, podemos escribir:

$$\phi_i^{n+1} = \phi_i^n - \Delta t (\max(D_i^{-x}\phi, 0)^2 + \min(D_i^{+x}\phi, 0)^2))^{1/2} \quad (1.61)$$

Este es el esquema Level Set dado en [11].

Estos esquemas son explícitos con respecto al tiempo y por lo tanto pueden ser programados en una manera acertada. La resolución de este esquema en todo el dominio computacional es conocida como “acercamiento matricial completo”, dado que se actualiza todo el Level Set en cada paso de tiempo, no simplemente el Level Set cero correspondiente al frente.

Problema de valor de contorno: el método estacionario.

La formulación del mismo es:

$$|\nabla T|F(x, y) = 1 \text{ tal que } \Gamma = \{(x, y) \mid T(x, y) = 0\}$$

donde Γ es la posición inicial de la interfaz. Se genera entonces el siguiente esquema convexo de primer orden:

$$\left[\begin{array}{l} \max(D_{ijk}^{-x}T, 0)^2 + \min(D_{ijk}^{+x}T, 0)^2 + \\ \max(D_{ijk}^{-y}T, 0)^2 + \min(D_{ijk}^{+y}T, 0)^2 + \\ \max(D_{ijk}^{-z}T, 0)^2 + \min(D_{ijk}^{+z}T, 0)^2 \end{array} \right]^{1/2} = \frac{1}{F_{ijk}} \quad (1.62)$$

Este esquema es una ecuación no lineal para los valores de la cuadrícula T_{ijk} . Una solución óptima a este esquema es a través del método Fast Marching, que es discutido a continuación.

El método Fast Marching.

Los métodos estándar para resolver el problema de valor límite precisan iteración, sin embargo el método Fast Marching permite solucionar este problema sin iteración, para lo que se produce un ordenamiento óptimo de los puntos de la malla tal que se pueda resolver el sistema no-lineal con una sola aplicación del mismo y una sola iteración. La ecuación (1.62) sufre un ligero cambio y queda de una forma más conveniente:

$$\left[\begin{array}{l} \max(D_{ijk}^{-x}T, -D_{ijk}^{+x}T, 0)^2 + \\ \max(D_{ijk}^{-y}T, -D_{ijk}^{+y}T, 0)^2 + \\ \max(D_{ijk}^{-z}T, -D_{ijk}^{+z}T, 0)^2 \end{array} \right]^{1/2} = \frac{1}{F_{ijk}} \quad (1.63)$$

Para solucionar la ecuación (1.63) se puede iterar en todo el dominio, partiendo de las condiciones iniciales. Para el punto T de la cuadrícula y sus cuatro vecinos TA, TB, TC, TD (ver Figura 1.16) se actualiza el valor de T en cada uno de estos puntos, según la ecuación de segundo grado (1.63) hasta que una solución sea alcanzada. La ecuación cuadrática se resuelve para $T_{ij}^{nível+1}$, dados $T_{i-1j}^{nível}, T_{i+1j}^{nível}, T_{ij-1}^{nível}, T_{ij+1}^{nível}$, donde $nível = 1 \dots N$ e $i, j = 1 \dots \dim$.

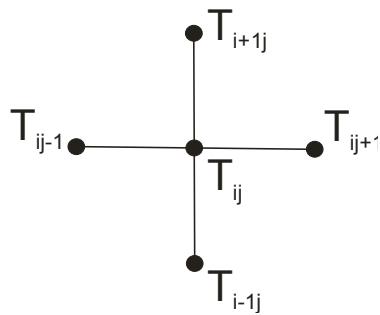


Figura 1.16. Actualización de un punto de la cuadrícula.

Para mejorar el tiempo de ejecución del algoritmo la idea central del método Fast Marching es construir sistemáticamente la solución T usando solamente valores en sentido contrario a la propagación, es decir, se escogen los menores valores de T del exterior, dando un paso fuera de la condición del límite “a favor” de la propagación. Se recorre la parte delantera del frente considerando sólo aquellos puntos que se encuentran a corta distancia del mismo, creando así una banda, sobre la que se congelan los valores de los puntos existentes y se incorporan los nuevos puntos. La clave está en seleccionar qué punto de la cuadrícula en la banda se actualizará.

En la siguiente figura el valor conocido del frente se muestra como un punto negro. Los puntos grises son puntos de la cuadrícula donde el valor de solución es desconocido.

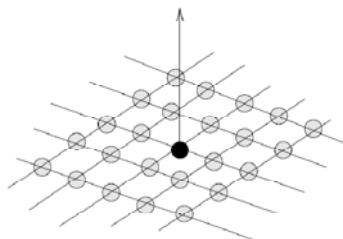


Figura 1.17. Comienzo del método Fast Marching.

Se inicia el algoritmo solucionando la ecuación (1.62) dada las condiciones iniciales, calculando valores nuevos en cada uno de los cuatro puntos vecinos de la cuadrícula, como se muestra en la Figura 1.18. Esto provee posibles valores para T en cada punto de la cuadrícula $T_{i-1,0}, T_{i+1,0}, T_{0,j-1}, T_{0,j+1}$, ilustrados como puntos grises oscuros en la figura. De estos puntos se selecciona el punto gris oscuro con menor valor como el próximo valor de T , puesto que el método no permite que ningún punto pueda ser

afectado por puntos de la cuadrícula que contienen mayores valores que T . El nuevo punto se congela y se continúa con el algoritmo, como se muestra en la Figura 1.19.

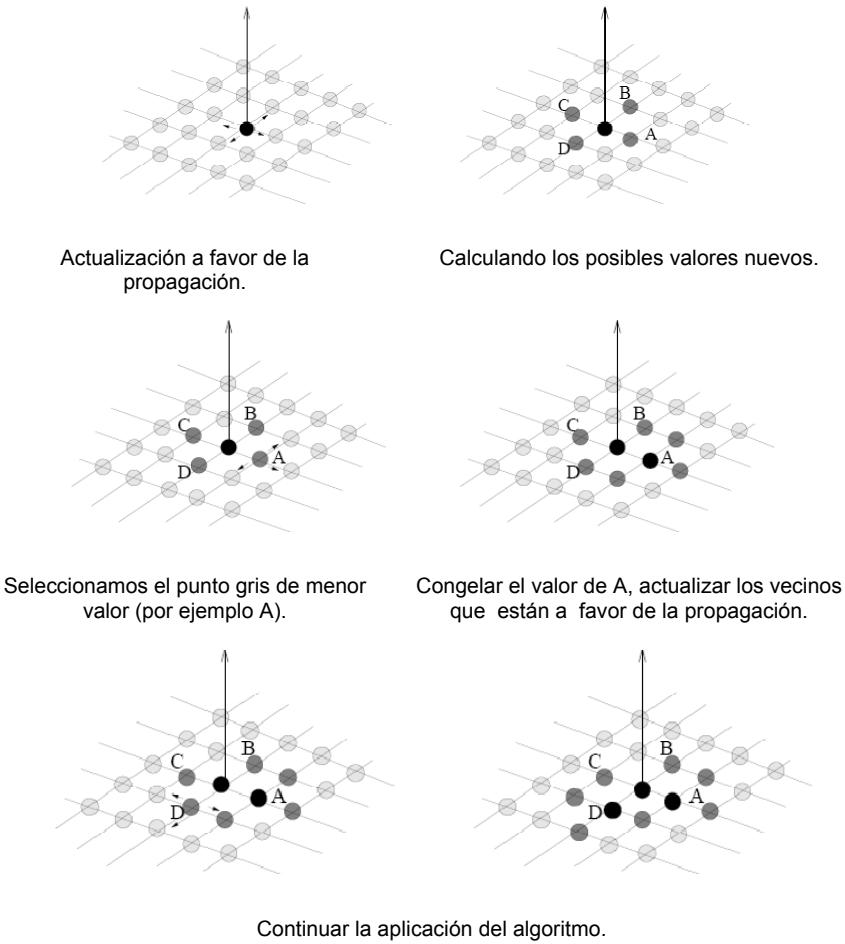


Figura 1.18. Actualizaciones producidas por el método Fast Marching.

Al recalcular los valores de los puntos vecinos de T que se encuentran a favor de la dirección de propagación, no pueden generarse valores menores que los de cualquier punto ya conocido, por lo que se puede avanzar tomando siempre el punto con menor valor considerándolo como el próximo valor conocido, y actualizar sus vecinos (Figura 1.19). Los valores conocidos no cambian para los cálculos posteriores.

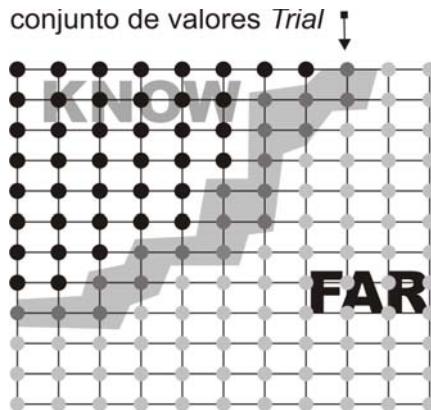


Figura 1.19. Progreso del método Fast Marching.

1.3.4 Segmentación usando los Level Set.

Para emplear los métodos de evolución de contornos en este caso en particular se propone emplear para el Fast Marching la función de velocidad F brindada por Sethian en [9]:

$$F(x, y) = g_l = e^{-\alpha|\nabla(G_\sigma * I(x, y))|}, \alpha > 0 \quad (1.64)$$

Donde la expresión $G_\sigma * I$ denota la convolución de la imagen con un filtro de suavizado gausiano cuya anchura característica es σ . El término $\nabla(G_\sigma * I(x, y))$ es en esencia cero excepto donde la imagen cambia repentinamente el gradiente, en este caso el valor de la expansión es grande. Así el filtro, $g_l(x, y)$ está próximo a la unidad fuera de los límites y tiende a cero cerca de los cambios bruscos en el gradiente de la imagen. Estos cambios probablemente corresponden a los bordes de la forma deseada. En otras palabras, la función del filtro anticipa las caídas profundas del gradiente de la imagen y retarda la evolución del frente pasando fuera de la región deseada.

Para el método Level Set se propone emplear la función de velocidad definida en [9] y que plantea:

$$F(x) = g_l(x) = \frac{1}{1 + |\nabla G_\sigma * I(x)|} \quad (1.65)$$

Se utiliza además para la investigación una metodología de trabajo propuesta en [9,11] que propone la combinación de los métodos Level Set y Fast Marching en pos de buscar velocidad y precisión en la segmentación de la imagen, que plantea primeramente aplicar el método Fast Marching para resolver la siguiente ecuación (1.63) hasta que la solución casi haya alcanzado el borde deseado; esto ocurre donde la solución T toma valores altos, y por lo tanto el flujo casi se ha detenido. Esto provee las condiciones iniciales para el método Level Set, buscando mayor precisión en la segmentación.

1.4 Análisis de forma.

Con los contornos obtenidos empleando la segmentación realizada por los métodos propuestos se hace un análisis morfológico, para determinar la cantidad de eritrocitos existentes que están deformados. Este análisis se realiza obteniendo medidas sobre los contornos finales, entre las que se tienen el área y el perímetro. Otras empleadas fueron [1]:

Coeficiente de compacidad (CSF).

Es uno de los parámetros geométricos utilizados en la comparación de objetos que son observados desde diferentes distancias, ya que el mismo no depende del tamaño del objeto. La compacidad es un número definido como:

$$F = \frac{4\pi A}{P^2} \quad (1.66)$$

El valor de compacidad será mínimo para objetos circulares, pero esta medida es sensible al ruido. Objetos con bordes ruidosos tendrán un perímetro muy largo en comparación con objetos que tienen bordes suaves.

Coeficiente elíptico (ESF).

Es una medida que expresa la elongación del objeto. Para obtenerla se tiene en cuenta la razón entre las abscisas mayor y menor del objeto, que es 1 si el objeto tiene forma circular.

Para determinar si las células segmentadas están deformadas se emplearon los valores determinados en [12] para los coeficientes de compacidad y elíptico, donde se considera que si el $ESF < 0.5$ la célula es deformada alargada, para valores de $CSF < 0.8$ y $ESF > 0.5$ la célula es deformada poco alargada, o sea, tiene otra deformación, y si el valor de $CSF > 0.8$ y el $ESF > 0.5$ la célula se considera discoidal normal, es decir, tiene la forma circular característica de los eritrocitos.

1.5 Conclusiones parciales.

En este capítulo se explicaron los detalles de los distintos métodos referentes a análisis de texturas empleados, que son el momento de orden k , la matriz de co-ocurrencia y los modelos de Gabor, pertenecientes a las clases estadística y espectral respectivamente, así como las combinaciones de los tres métodos.

Se establecieron los criterios de comparación entre vectores característicos y se detallaron los algoritmos de agrupamiento a emplear para obtener una división por regiones en una imagen según los vectores de características obtenidos en la misma, que fueron el algoritmo K-means y el β_0 -conexo.

Se explicó la estrategia a seguir para la detección de regiones de interés iniciales, la creación de una base de conocimiento o matriz de aprendizaje y la clasificación a partir del conocimiento adquirido mediante los algoritmos k-NN y votación.

Se detallaron los métodos de segmentación empleados, basados en la evolución de contornos, específicamente los solucionados empleando los métodos Level Set tanto dependientes del tiempo como estacionarios.

Por último se realizó un análisis de las medidas utilizadas para determinar la forma de un objeto entre las que se encuentran área, perímetro y los coeficientes elíptico y de compacidad.

CAPÍTULO 2. DETALLES DE IMPLEMENTACIÓN.

2.1 Introducción.

En este capítulo se discutirán los detalles de implementación de los algoritmos citados en el capítulo anterior, además de la metodología a seguir para la obtención de los objetivos planteados, específicamente para la detección, segmentación y análisis morfológico de los objetos de interés en las imágenes. Se exponen los pseudocódigos de los algoritmos implementados. Se detallan además las principales funcionalidades de un prototipo de software que permite el uso y análisis de estos algoritmos para el problema en estudio. La implementación del sistema se realizó en la plataforma de desarrollo MatLAB por las potencialidades de la misma para el procesamiento de imágenes.

El desarrollo del proceso se dividió en dos fases, la primera llamada fase de entrenamiento dirigida a la creación y entrenamiento de la matriz de aprendizaje para la detección de los objetos de interés a través de uno de los algoritmos de clasificación descritos y la segunda llamada fase de análisis en la cual se efectúa la detección inicial y segmentación de los eritrocitos y el posterior análisis morfológico de los bordes detectados para efectuar el conteo de los que se hallan deformados. La Figura 2.1 muestra un esquema en el cual se describen ambas fases.

La obtención de la matriz de aprendizaje en la primera fase es un proceso cíclico que se realiza hasta que se considere que se ha adquirido el conocimiento necesario sobre las texturas de interés.

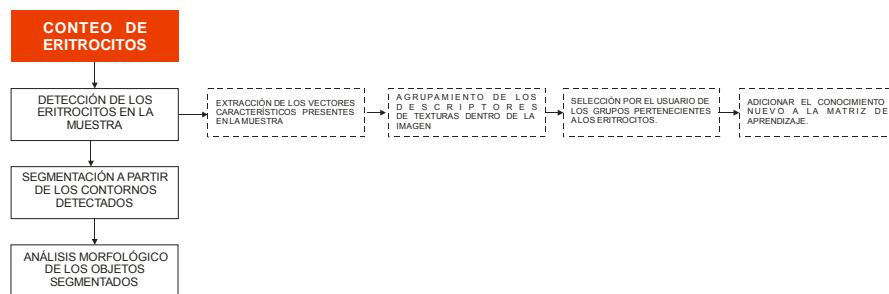


Figura 2.1. Proceso para el conteo de eritrocitos.

2.2 Extracción de características.

Para la extracción de características se crearon una serie de estructuras con el fin de dotar al sistema de una manera sencilla de incorporar nuevos métodos de extracción. Entre las estructuras definidas tenemos las siguientes:

```
typedef struct txCARACT{ // métodos definidos

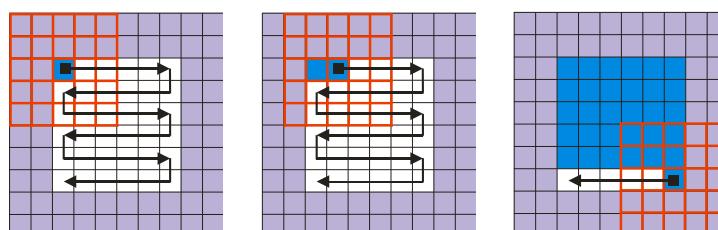
    NOMBREMETODO NombreMetodo1;
    NOMBREMETODO NombreMetodo2;
    ...
    NOMBREMETODO NombreMetodoK;
} CARACT;

typedef struct txNOMBREMETODO { // información de cada método

    handle hfunc;           // dirección del método
    char* strmedidas;       // nombre de los rasgos
    int index;               // índice de las medidas en Ti

} NOMBREMETODO;
```

Para aplicar el método de extracción de características seleccionado se debe crear una instancia de la estructura *CARACT* a partir de la función *createstructcaract* que recibe como parámetro el nombre de dicho método, la cual será empleada por el algoritmo *wndpass* para la extracción de las características en toda la imagen. Este algoritmo aplica los métodos de extracción realizando un barrido de la ventana de observación por la imagen (Figura 2.2), en cada paso se extraen las características obteniéndose el vector *T* que se asocia al píxel central de la ventana.



a) Paso 1

b) Paso 2

c) Paso n-ésimo

Figura 2.2. Barrido de la ventana de observación.

El algoritmo propuesto para realizar este proceso es el siguiente:

Algoritmo wndpass.

Entrada:

- I imagen a la que se le realizará la extracción.
- h tamaño de la ventana de observación.
- $caract$ métodos de extracción a utilizar.

Salida:

- X una lista con todos los vectores característicos.

Método:

Paso 1: Añadir celdas artificiales a la imagen.

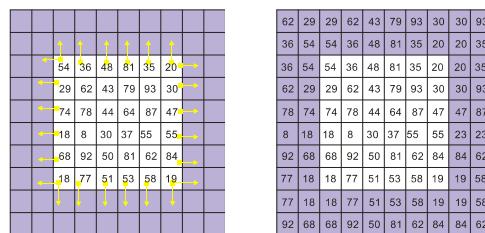
Paso 2: Colocar la ventana de observación en la primera posición de la imagen (ver Figura 2.2 (a)).

Paso 3: Calcular las medidas por cada uno de los métodos especificados en *namecaract* e insertarlos en $X(i)$ para la i -ésima iteración (en este paso se emplea la estructura CARACT que permite la extracción del vector característico).

Paso 4: Mover la ventana a la próxima posición según el recorrido propuesto en la Figura 2.2.

Paso 5: Ir al paso 3 hasta que se hallan calculado todos los vectores asociados a la imagen original.

La ventana de observación no está definida completamente en los bordes de la imagen, por lo que se requiere añadir celdas artificiales a la imagen en el paso 1 para poder calcular los vectores característicos en los bordes (ver en el Capítulo 1 epígrafe 1.2.1).



a) Expansión. b) Celdas artificiales

Figura 2.3. Imagen a la que se le añaden celdas artificiales.

Para ello se empleó la función *padarray* de MatLAB, la Figura 2.3 (b) muestra el resultado de su aplicación para una matriz de 6×6 y su sintaxis es la siguiente:

B = padarray(A, padsiz, padval, direction).

donde

- A : es la imagen de entrada.
- $padsiz$: la cantidad de filas y columnas que se van a adicionar en cada dirección, en este caso el valor está relacionado con el tamaño de la ventana de observación, $padsiz=h/2$ donde h es el tamaño de la ventana.

- *padval*: especifica cómo se obtendrán los valores de las celdas añadidas, en *padval='symmetric'* los valores son obtenidos como se muestran en la Figura 2.3 (a).
- *direction*: es la dirección en que serán añadidas las nuevas celdas, para *direction='both'* las celdas se añadirán a ambos lados de cada dirección.

A continuación se detalla lo relacionado con la implementación de los tres métodos de extracción de características empleados.

2.2.1 Momento de orden k.

Para la implementación de una función que extraiga las medidas estadísticas basadas en el momento de orden k, se hace uso de la versión propuesta en [4]:

$T = \text{statxture}(I, scale)$,

donde

- *I*: imagen a la que se le extraerán las características.
- *scale*: la cantidad de medidas a extraer (por defecto todas).

En la Tabla 2.1 se muestran los resultados de esta función para tres texturas que se muestran en la Figura 2.4.

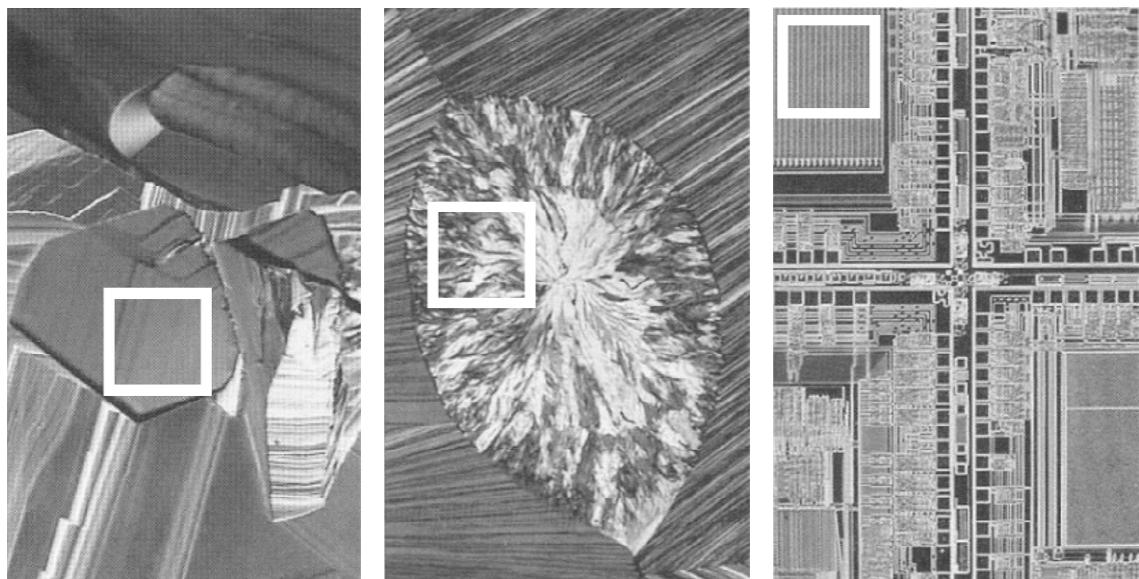


Figura 2.4. Tres tipos de texturas.

Tabla 2.1. Ejemplo de tres vectores característicos.

Textura	Media	Desviación Estándar	Suavidad	Tercer Momento	Uniformidad	Entropía
Suave	82.64	11.79	0.002	-0.105	0.026	5.434
Rugosa	143.56	74.63	0.079	-0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

2.2.2 Matriz de co-ocurrencia.

Para el cálculo de la matriz de co-ocurrencia se empleó la función de MATLAB *graycomatrix*, que tiene la siguiente sintaxis:

```
glcm = graycomatrix(I, 'NumLevels', numLevel, 'G', [], 'Offset', offset, 'Symmetric, true'),
```

donde

- *I*: imagen a procesar.
- *numlevel*: número de niveles de grises que se va a utilizar en la construcción de la matriz de co-ocurrencia, este es un valor entero (por defecto es 8 y si la imagen es binaria es 2).
- '*G*' o '*GrayLimits*': especifica cómo son escalados los valores de la imagen para obtener la matriz de co-ocurrencia, emplea el elemento [*LOW HIGH*], si '*GrayLimits*' es [] (por defecto), se utilizan [1; *NumLevels*].
- '*offset*': arreglo de *p* filas (cantidad de ángulos para determinar las matrices de co-ocurrencia) y 2 columnas, indica distancia y ángulo para determinar la relación entre píxeles. Cada fila del arreglo es un vector de dos elementos [*FILA COLUMNA*], que serán el número de filas y columnas entre los píxeles respectivamente, son valores enteros y por defecto se asume [0 1]. La Tabla 2.2 muestra la equivalencia entre los ángulos y [*FILA COLUMNA*].

Tabla 2.2. Relación de los ángulos con el 'offset'.

Ángulo	Offset
0°	[0 d]
45°	[-d d]
90°	[-d 0]
135°	[-d d]

La Figura 2.5 muestra el cálculo de la matriz de co-ocurrencia para una distancia *d*=1 y ángulo 0° en una imagen con tres niveles de gris, *z*₁=0, *z*₂=1 y *z*₃=2. En la matriz

resultante A la posición (0,0) indica que en la imagen existen 4 ceros a la derecha de otro cero.

$$\begin{matrix} 0 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{matrix} \quad A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

a) Imagen original b) Matriz de co-ocurrencia

Figura 2.5. Cálculo de la matriz de co-ocurrencia.

Por esta función se obtendrán 4 matrices de co-ocurrencia (una para cada ángulo) a las que se le extraerán las medidas descritas en el Capítulo 1 epígrafe 1.2.1 para obtener 4 vectores característicos por cada una de ellas y el promedio de los mismos es el vector resultante T de este proceso.

2.2.3 Modelos de Gabor.

Para aplicar este método se crearon ocho de filtros de Gabor empleando la función propuesta en [14] *gabfiltros*. Los mismos se muestran en la Figura 2.6.

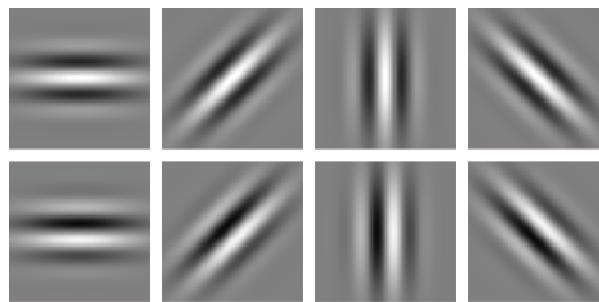


Figura 2.6. Banco de filtros de Gabor utilizados.

El vector resultante tendrá veinticuatro elementos que es el resultado de extraerle las 3 medidas descritas en el Capítulo 1 epígrafe 1.2.1 a las ocho convoluciones realizadas a la imagen con cada uno de los filtros. La función utilizada para la convolución de la imagen fue la que ofrece MatLAB, *imfilter* que presenta la siguiente sintaxis:

B = imfilter(A,H,option),

donde

- A : imagen o arreglo multidimensional.
- H : filtro.
- $option$: el valor de este parámetro para la convolución debe ser igual a 'conv'.

Dadas estas funciones el cálculo del vector característico por el método de Gabor se realiza a partir del siguiente algoritmo.

Algoritmo gabvector.
Entrada:
<ul style="list-style-type: none"> ▪ wnd ventana de observación. ▪ h tamaño de la ventana.
Salida:
<ul style="list-style-type: none"> ▪ T vector característico.
Método:
Paso 1: Calcular el banco de filtros de Gabor dado el tamaño de la ventana h a partir de la función <i>gabfiltros</i> . Paso 2: Para cada uno de los filtros obtenidos en el paso anterior hacer: <ol style="list-style-type: none"> Calcular la convolución de la ventana <i>wnd</i> con el filtro i-ésimo. Calcular cada una de las medidas descritas en el Capítulo 1 epígrafe 1.2.1 para la ventana convolucionada en el paso anterior y asignarla al vector T en la posición i.

2.3 Agrupamiento de texturas.

Para el agrupamiento de las texturas de la imagen se implementaron los algoritmos propuestos en el Capítulo 1 epígrafe 1.2.2 K-means y β_0 -conexo, que serán empleados indistintamente en dos estrategias propuestas para la creación de la matriz de aprendizaje descritas en el Capítulo 3 epígrafe 3.4.

El algoritmo K-means se basa en la distancia de los objetos en el espacio, en el cual se colocan k centroides en las aglomeraciones de vectores en el espacio n-dimensional. Un ejemplo de los grupos formados por este algoritmo para vectores en dos dimensiones se muestra en la Figura 2.7 y el seudocódigo para el mismo se muestra más adelante.

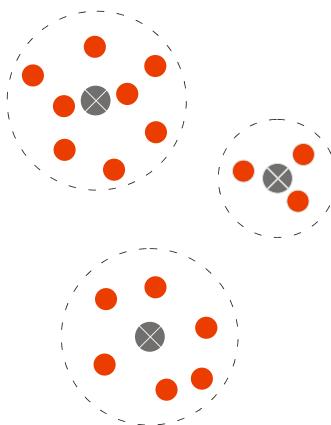


Figura 2.7. Grupos formados por el algoritmo K-means para dos dimensiones.

Algoritmo K-means.

Entrada:

- O conjunto de vectores característicos de las texturas.
- k número de grupos.

Salida:

- G vector que indica el grupo de cada objeto en O .
- C centroides de cada grupo.

Método:

Paso 1: Inicializar los k primeros centroides aleatoriamente a partir de los objetos del conjunto O y $oldG$ en vacío.

Paso 2: Hacer $oldG$ igual a G .

Paso 3: Calcular la distancia de cada objeto a cada centroide en C y asignarlo al grupo del centroide con el cual obtuvo la menor distancia.

Paso 4: Si $oldG$ es distinto de G recalcular el centoride $C(i)$ para cada grupo i e ir al paso 2. Sino terminar.

Este algoritmo por las formas que adquieren los grupos que él crea lo hace incapaz de en algunos casos de diferenciar objetos con texturas un tanto similares, por lo que se valoró el uso del algoritmo de clasificación β_0 -conexo que supone la creación de una matriz de semejanza en la que sus dimensiones dependen de la cantidad de objetos de la muestra. Para una imagen de dimensiones $n \times m$ la matriz de semejanza asociada será de dimensiones $(n \times m) \times (n \times m)$ que es un número considerablemente alto. Esta matriz puede ser reducida para trabajar solamente con los elementos de su diagonal superior o inferior a $(n \times m) \times (n \times m - 1)/2$ elementos. Por ejemplo para una imagen de 1000x750 píxeles la cantidad de elementos de la matriz de semejanza serían de 562 500 000 000, o en caso del uso de su diagonal superior de 281 249 625 000. Si este número se multiplica por el tamaño que ocupa la representación del tipo de los

elementos de la matriz en memoria (por ejemplo 8-bit para un entero en MatLAB) se obtendría el espacio requerido para almacenar esta matriz. En el caso particular de MatLAB el máximo de memoria posible de manera contigua para un arreglo es de 606 700 000 bytes, que como se puede observar es menor que los 281 249 625 000 bytes necesarios para almacenar los elementos de la imagen del ejemplo anterior.

En este trabajo se propone modificar el cálculo de la matriz de aprendizaje para obtener en cada paso del algoritmo la fila de esta matriz que será empleada, por lo que no es necesario cargarla completamente en la memoria.

Algoritmo β_0 -conexo-modificado.

Entrada:

- X vector de rasgos característicos asociado a cada píxel de la imagen.
- β_0 umbral de semejanza.

Salida:

- G grupos de cada objeto en X .

Método:

Paso 1: Inicialmente cada objeto es un grupo unitario.

Paso 2: Calcular la semejanza de $X(i)$ en la iteración i con los demás objetos a partir del $X(i+1)$.

Paso 3: Seleccionar los objetos β_0 -semejante a $X(i)$.

Paso 4: Unir los grupos de los objetos obtenidos en el paso anterior al grupo de $X(i)$.

Paso 5: Mientras i sea distinta a la cantidad de elementos de X , ir al paso 2.

Para este caso se usó la función de semejanza dada en la ecuación 1.31, con la que se obtuvieron los mejores resultados para el algoritmo.

La obtención de los representantes para los grupos extraídos por el algoritmo β_0 -conexo no es trivial (ver Capítulo 1 epígrafe 1.2.2) debido a que estos grupos no son circulares como en el caso del algoritmo K-means y por tanto el objeto promedio no satisface la condición de representante. En este trabajo se propone la implementación de un algoritmo para el cálculo de los representantes de los grupos obtenidos por el algoritmo β_0 -conexo. La idea central de este algoritmo es ir agrupando los objetos en cada grafo de β_0 -semejanza que estén a una distancia β_0 de un centroide. La Figura 2.8 muestra este proceso y más adelante se detalla el algoritmo.

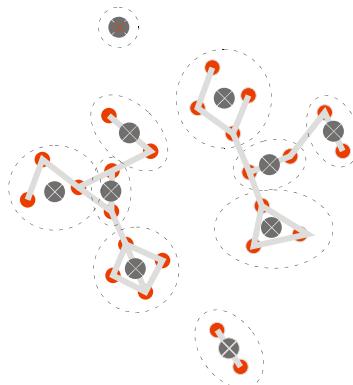


Figura 2.8. Cálculo de los representantes del algoritmo β_0 -conexo.

Algoritmo para el cálculo de los representantes del β_0 -conexo.	
Entrada:	<ul style="list-style-type: none"> ▪ O vector de rasgos característicos asociado a cada píxel de la imagen. ▪ β_0 umbral de semejanza.
Salida:	<ul style="list-style-type: none"> ▪ R vector representante del grupo.
Método:	<p>Paso 1: Si O tiene un solo elemento, hacer a O el representante del grupo y terminar.</p> <p>Paso 2: Hacer $R(1)$ igual a $O(1)$ y marcar $O(1)$ visitado por $R(1)$.</p> <p>Paso 3: Calcular la distancia del objeto $O(i)$ en la iteración i a cada uno de los representantes.</p> <p>Paso 4: Seleccionar el representante $R(p)$ de menor distancia a $X(i)$.</p> <p>Paso 5: Si no existen crear un nuevo representante con el valor de $X(i)$ y marcarlo como visitado por $R(i)$. Sino marcar a $X(i)$ como visitado por $R(p)$ y hacer $R(p)$ igual al promedio de los elementos visitados por él.</p> <p>Paso 6: Ir al paso 3 hasta llegar al objeto $O(n)$.</p>

2.4 Creación de la matriz de aprendizaje.

El proceso de obtención de la matriz de aprendizaje es cíclico (ver Capítulo 1 epígrafe 1.2.3), se van adquiriendo matrices de aprendizaje a partir de los métodos descritos y se van uniendo en cada paso empleando el siguiente algoritmo.

Algoritmo UnirMatriz.

Entrada:

- MA matriz de aprendizaje.
- K clases correspondientes a cada objeto de MA .
- O objetos a añadir en MA .
- C clase de los objetos de O .
- t umbral de tipicidad.

Salida:

- MA^* matriz resultante de la unión.
- K^* clases de MA^* .

Método:

Paso 1: Calcular la distancia del objeto $O(i)$ a cada uno de los objetos de MA en la iteración i .

Paso 2: Seleccionar la menor distancia d_j al objeto j perteneciente a MA .

Paso 3: Si d_j es menor que un épsilon y $K(i)$ es distinta a $C(j)$ (Esto es si las clases de cada objeto j e i son diferentes) hacer $K(i)$ igual $C(j)$ e ir al paso 1.

Paso 4: Calcular el coeficiente de tipicidad del objeto $O(i)$ con respecto a los objetos de MA de su misma clase a partir de las distancias calculadas en el paso 1.

Paso 5: Si el coeficiente de tipicidad del objeto $O(i)$ es menor que t agregarlo a MA e ir al paso 1.

Paso 6: Hacer MA^* igual a MA y K^* igual a K . Terminar.

El coeficiente de tipicidad del objeto O_i mide la representatividad del objeto en el grupo y se define a través de la siguiente expresión [7]:

$$t(O_i) = \frac{\beta(O_i)}{\frac{1}{G-1} \sum_{j=1}^G (\bar{\beta}(O_i) - \beta(O_i, O_j))^2}$$

(2.1)

donde

$$\bar{\beta}(O_i) = \frac{1}{G-1} \sum_{j=1}^G \beta(O_i, O_j) \quad (2.2)$$

para $O_i \in Grupo(O_i)$ y $G = |Grupo(O_i)|$

Las matrices después de ser unidas deben reducirse para garantizar que la resultante no alcance dimensiones inadecuadas (ver Capítulo 1 epígrafe 1.2.3). El algoritmo para esto es como sigue.

Algoritmo ReducirMatriz.

Entrada:

- MA matriz de aprendizaje.
- K clases de la matriz de aprendizaje MA .

Salida:

- MA^* matriz de aprendizaje reducida.
- K^* clases de MA^* .

Método:

- Paso 1:** Calcular el umbral de semejanza β_0 para los objetos de la clase K en la i -ésima iteración.
- Paso 2:** Agrupar los objetos de la clase K de MA a partir del algoritmo de clasificación β_0 -compacto.
- Paso 3:** Para cada objeto perteneciente a cada uno de los grupos obtenidos en el paso anterior calcular el coeficiente de tipicidad.
- Paso 4:** Seleccionar por grupo el objeto de mayor tipicidad y agregarlo a su respectiva clase en MA^* .
- Paso 5:** Ir al paso 1 hasta que i sea igual al número de clases de MA .

2.5 Detección inicial de objetos.

Para la detección de objetos de interés en la imagen a partir de su textura, se emplearon los algoritmos de clasificación k-NN y votación. Un objeto en la imagen se considera detectado si los algoritmos de clasificación logran catalogar una región o al menos un píxel del objeto como perteneciente a las texturas de interés definidas en el proceso de creación de la matriz de aprendizaje. Para este trabajo se definieron dos clases: texturas de interés para el proceso (OI) y textura que no lo son (ONI). Estas clases no están solapadas, pues los vectores característicos no pueden considerarse OI y ONI al mismo tiempo. Ambos algoritmos se muestran a continuación.

Algoritmo k-NN.

Entrada:

- MA objetos de la matriz de aprendizaje.
- K clases de MA .
- k cantidad de vecinos a votar.
- O objetos que se desean clasificar.

Salida:

- C clases asociadas a los elementos de O .

Método:

- Paso 1:** Calcular la semejanza de $O(i)$ en la iteración i -ésima con cada objeto de la matriz de aprendizaje.
- Paso 2:** Seleccionar los k objetos de MA que más se parecen a $O(i)$.
- Paso 3:** Asignarle a $C(i)$ la clase predominante entre los objetos seleccionados en el paso anterior.
- Paso 4:** Mientras i sea menor o igual que m , donde m es la cantidad de elementos de MA ir al paso 1.

Algoritmo de votación.

Entrada:

- MA objetos de la matriz de aprendizaje.
- K clases de MA .
- O objetos que se desean clasificar.

Salida:

- C clases asociadas a los elementos de O .

Método:

Paso 1: Calcular el voto Γ_w que le otorga MA a $O(i)$ en el paso i -ésimo para cada uno de los conjuntos de apoyo w a través de la siguiente ecuación:

$$\Gamma_w(O(i), MA(j)) = \beta(I(wO(i)), I(wMA(j))), \quad j=1 \dots m$$

donde

wO rasgos o medidas de O pertenecientes al conjunto de apoyo w .

m cantidad de objetos de MA .

Paso 2: Calcular a partir de los votos obtenidos en el paso anterior Γ_w el voto Γ_w^j de cada clase al objeto $O(i)$ en cada uno de los conjuntos de apoyo:

$$\Gamma_w^j(O) = \frac{1}{|K_j|} \sum_{O_i \in K_j} \Gamma_w(O, O_i).$$

Paso 3: Calcular el voto $\Gamma_j(O)$ de cada clase al objeto $O(i)$ a partir de Γ_w^j :

$$\Gamma_j(O) = \frac{1}{|\Omega|} \sum_{w \in \Omega} \Gamma_w^j(O)$$

Ω conjunto formado por todos los conjuntos de apoyo.

Paso 4: Asignar a $C(i)$ la clase de mayor cantidad de votos en Γ_j .

2.6 Segmentación a partir de los métodos Level Set.

A partir del proceso anterior se obtienen puntos o regiones pertenecientes a los objetos de interés, los que serán utilizados como puntos semillas para la inicialización de los algoritmos de segmentación descritos en el Capítulo 1 epígrafe 1.3.4. Este proceso se muestra en la Figura 2.9. Para este trabajo se implementaron los algoritmos Fast Marching, Level Set y la combinación Fast Marching Level Set.

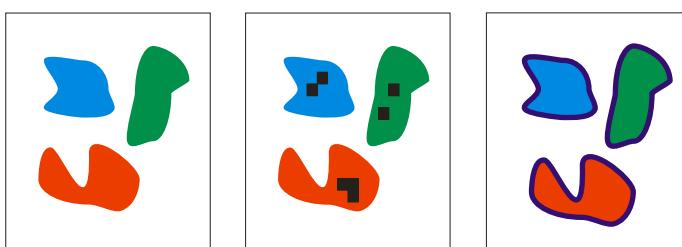


Figura 2.9. Proceso de segmentación.

2.6.1 Algoritmo Level Set. Acercamiento matricial completo.

El algoritmo Level Set requiere de condiciones iniciales, que serán obtenidas a partir de las regiones detectadas empleando las características de texturas valoradas anteriormente. La inicialización exige construir una función Level Set tal que al ser evaluada en los valores de los contornos de las regiones iniciales esta función sea igual a cero. En el resto de los puntos de la malla definida para la función Level Set los valores serán igual a la distancia de cada cuadrícula al frente, tomando valores negativos en el interior de la región definida y positivos en el exterior. Para construir esta función se emplea el Fast Marching con velocidad del frente de propagación $F=1$, que es la vía más eficiente propuesta en [9] para realizar esta construcción. La Figura 2.10 muestra una inicialización por este método.

5	4	3	2	2	2	2	3	4	5
4	3	2	1	1	1	1	2	3	4
3	2	1	0	0	0	0	1	2	3
2	1	0	0	-1	-1	0	0	1	2
2	1	0	-1	-2	-2	-1	0	1	2
2	1	0	-1	-2	-2	-1	0	1	2
2	1	0	0	-1	-1	0	0	1	2
3	2	1	0	0	0	0	1	2	3
4	3	2	1	1	1	1	2	3	4
5	4	3	2	2	2	2	3	4	5

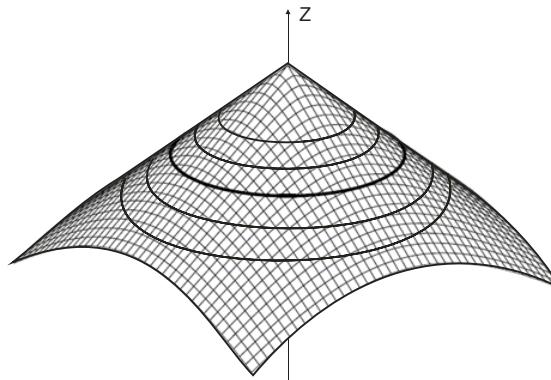


Figura 2.10. Función de distancia signed.

Algoritmo Level Set.

Entrada:

- *Data* inicialización para el Level Set.
- *Maxiter* máximo de iteraciones.
- *F* la velocidad de la expansión definida para el dominio de la imagen.
- *Eps* umbral para la variación de *T* en la banda.

Salida:

- *Data* cuadrícula con los contornos finales.

Método:

Paso 1: Inicializar *Y* igual *Data*.

Paso 2: Calcular los valores para el gradiente *Grd* según la ecuación 1.61.

Paso 3: Hacer *Yold* igual *Y*.

Paso 4: Actualizar la función Level Set dada en la ecuación 1.61.

$$Y = Y + \delta T * Grd.$$

Paso 5: Comparar los valores de *Y* con los de *Yold*, si el promedio de las diferencias de cada uno de ellos para cada posición no supera un cierto umbral *Eps*, terminar.

2.6.2 Algoritmo Fast Marching.

Este algoritmo también requiere de condiciones iniciales, que serán obtenidas, al igual que en Level Set, a partir de las regiones iniciales detectadas. Para la inicialización, se etiquetan todos los puntos de la cuadrícula como lejanos (*Far*) respecto al frente y se les asigna a los valores de T infinito (∞). Luego los valores de T pertenecientes al frente (*Known*) se igualan a cero y por último a los valores alrededor del frente (*Trial*) se les asigna $1/F$. La Figura 2.11 (a) muestra un ejemplo de las condiciones iniciales para el algoritmo Fast Marching.

∞	∞	1	∞	∞
∞	1	0	1	∞
1	0	1	0	1
∞	1	0	1	∞
∞	∞	1	∞	∞

a) Estructura data. b) Marcas para las celdas

Figura 2.11. Ejemplo de inicialización para el Fast Marching.

Para la implementación de este algoritmo se hizo uso de una estructura de cola de prioridades basada en un min-heap, que es un árbol binario completo con la propiedad de que el valor de cada uno de sus nodos es menor e igual que el de sus hijos. Una representación eficiente de este árbol es a través de un arreglo secuencial, en el cual se almacena un nodo en la posición k y sus hijos en la posición $2k$ y $2k+1$. De ahí que el padre de un nodo k se encuentre en la posición $k/2$. Por consiguiente el elemento menor de este arreglo es la raíz del árbol localizada en la posición $k=1$. En esta estructura se almacenan los valores de T etiquetados como *Trial* conjuntamente con su posición en la cuadrícula, para garantizar obtener el menor elemento del conjunto *Trial* en el menor tiempo posible. Para el uso y manipulación del árbol se implementaron varias funciones que serán comentadas a continuación.

La operación *BuilderGroup* dado un arreglo construye un min-heap, recorriendo el arreglo completo a partir del segundo elemento y moviendo cada uno hacia arriba o

hacia abajo (ver Figura 2.12) en busca de su posición correcta en cada paso del algoritmo, empleando las funciones *UpHeap* y *DownHeap*, respectivamente.

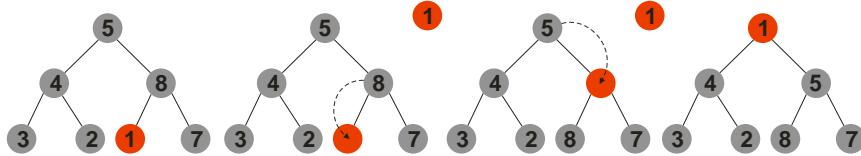


Figura 2.12. Moviendo el elemento de valor 1.

Luego el método Fast Marching se puede definir con el siguiente algoritmo.

Algoritmo Fast Marching.

Entrada:

- *Data* cuadrícula con las condiciones iniciales (ver Figura 2.11(a)).
- *Maxiter* máximo de iteraciones.
- *Eps* epsilon para los valores de *T*.

Salida:

- *Data* cuadrícula con los contornos finales.

Método:

- Paso 1:** A partir de los valores *Trial* de *Data* crear la estructura min-heap mediante la función *BuilderGrup* y asignarle a *oldT* el menor valor de *T*.
- Paso 2:** Asignarle a *A* la raíz del min-heap, eliminarla y realizar un barrido *DownHeap* para garantizar que el min-heap conserve sus propiedades.
- Paso 3:** Etiquetar el punto *A* como *Known*.
- Paso 4:** Etiquetar como *Trial* a todos los vecinos de *A* que no son *Known*. Si el vecino está en *Far*, entonces extraerlo y agregarlo al conjunto *Trial* en la estructura min-heap.
- Paso 5:** Recalcular los valores de *T* en *Data* para todos los vecinos *Trial* de *A* según la ecuación 1.63 y actualizar su valor en el min-heap. Esta última operación implica realizar un barrido *UpHeap* si el valor recalculado es menor que el anterior o un barrido *DownHeap* en caso contrario.
- Paso 6:** Si el número de iteraciones es igual a *Maxiter* o el módulo de la diferencia de *oldT* con *T* es menor que *Eps*, terminar.
- Paso 7:** Hacer a *oldT* igual *T*. Ir al paso 2.

El algoritmo Fast Marching es capaz de acercarse rápidamente a los bordes de los objetos de la imagen pero carece de cualidades deseables como lo es la suavidad. Además las condiciones de parada para el algoritmo no son ideales, para la segmentación de imágenes carecen de fortaleza pues debería conocerse a priori el número de iteraciones del algoritmo para alcanzar el borde deseado. Para atacar este problema en [9] se propone dejar iterar el algoritmo hasta que los puntos del frente

hayan alcanzado valores altos de T , lo que indicaría que el frente se ha acercado a los bordes. Esto todavía es una condición débil, puesto que habría que determinar qué valor de T alcanza el frente al acercarse a los bordes a segmentar y el porcentaje de puntos pertenecientes al frente que debieran poseer este valor.

En esta investigación se brinda una condición de parada para el algoritmo basada en la idea del uso y manipulación de la estructura utilizada para almacenar los valores de T . Es cierto que al frente acercarse a los bordes los valores de T sufren un cambio brusco, el problema es garantizar que realmente todos los puntos de T hayan sido afectados, y cuando esto suceda se podrá decir que se está lo suficientemente cerca del borde.

La estructura de datos que contiene los valores de la banda creada alrededor del frente y que es etiquetada como *Trial* en cada iteración del algoritmo extrae aquel de menor valor, lo que indica entre otras cosas en qué sentido avanzará la curva, haciendo que los valores que hayan sufrido el mayor cambio se vayan quedando atrás en la estructura hasta que los demás los alcancen. Esta es la esencia del algoritmo, ya que permite que la curva se desarrolle en las partes del frente de menor presión, y es también la idea básica de la condición de parada propuesta en la investigación. Cuando se extraiga de la estructura un valor de T que ha experimentado un cambio lo suficientemente brusco se puede garantizar que todos los valores T que quedan también lo han sufrido y la diferencia entre este valor y el que se extrajo en la iteración anterior es máxima. Luego bastaría chequear en cada iteración del algoritmo que esta diferencia supera un umbral determinado.

2.7 Análisis Morfológico.

Para el análisis morfológico se realizó una binarización de la imagen a partir de los contornos obtenidos por los algoritmos de segmentación empleados, con el objetivo de utilizar la función *regionprops* de MatLAB para obtener las medidas mencionadas en Capítulo 1 epígrafe 1.4 para cada una de las regiones obtenidas. Esta función presenta la siguiente sintaxis:

STATS = regionprops(L, properties),

donde

- *L*: es la imagen donde las regiones están etiquetadas con números diferentes.
- *Properties*: son las medidas que se desean extraer.
- *STATS*: *estructura* con las medidas extraídas para cada una de las regiones.

Luego el proceso de clasificación está descrito por el siguiente algoritmo.

Algoritmo para la clasificación de los contornos obtenidos.

Entrada

- *BW*: imagen binaria obtenida a partir de los contornos extraídos en la segmentación.

Salida:

- conteo de células (circulares, elongadas y otras deformaciones).

Método:

Paso 1: Etiquetar las regiones de *BW* obteniendo como resultado una nueva imagen *L*.

Paso 2: Calcular las medidas aplicando *STATS = regionprops(L, properties)*.

Paso 3: Calcular los coeficientes de compacidad (CSF) y elíptico (ESF) a partir de las medidas obtenidas en el paso anterior.

Paso 4: Se realiza el conteo de las células deformadas: si $ESF < 0.5$ la célula es elongada, si $CSF < 0.8$ y $ESF > 0.5$ la célula es deformada con otro tipo de deformación y $CSF > 0.8$ y el $ESF > 0.5$ la célula es discoidal normal que se corresponde con la forma circular característica de los eritrocitos.

Paso 5: Mostrar la gráfica correspondiente al conteo de células (circulares, elongadas y otras deformaciones).

2.8 Diseño del prototipo del sistema.

En este punto se detallan las funcionalidades de las interfaces correspondientes a cada una de las fases del sistema. Para la fase de entrenamiento se diseñó una interfaz que permite el empleo de los algoritmos y estrategias planteados, de igual manera la interfaz diseñada para la fase de análisis permite realizar el estudio de varias muestras de sangre para obtener el conteo de eritrocitos deformados en las mismas. La Figura 2.13 muestra ambas interfaces.



Figura 2.13. Interfaces del Proceso.

2.8.1 Interfaz del proceso de entrenamiento.

La interfaz correspondiente al proceso de entrenamiento permite:

- Extraer los vectores característicos a partir del método de extracción seleccionado por el usuario.
- Agrupar y extraer los representantes para cada grupo a partir de un algoritmo de agrupamiento seleccionado.
- Crear, unir y reducir las matrices de aprendizaje obtenidas a partir de la selección del especialista de los grupos de interés.

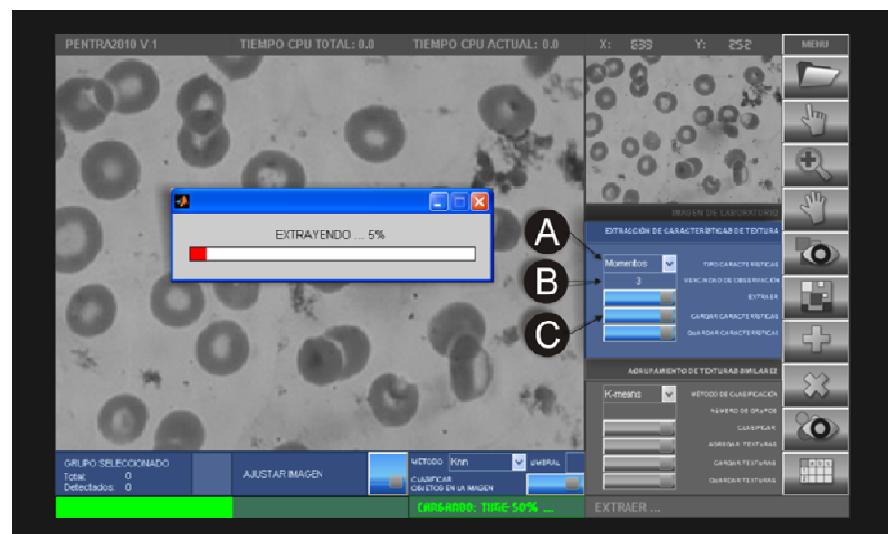


Figura 2.14. Proceso de extracción.

CAPÍTULO 2. DETALLES DE IMPLEMENTACIÓN.

Para la extracción de los vectores característicos el usuario selecciona en (A) el método de extracción a emplear y en (B) el tamaño de la ventana de observación para la cual se definirán las texturas que se obtendrán de la imagen. El sistema permite al usuario en (C) guardar y cargar las características que fueron extraídas.

El agrupamiento de los vectores característicos, requiere de la selección en (D) del algoritmo de clasificación a emplear además de los parámetros del mismo en (E).

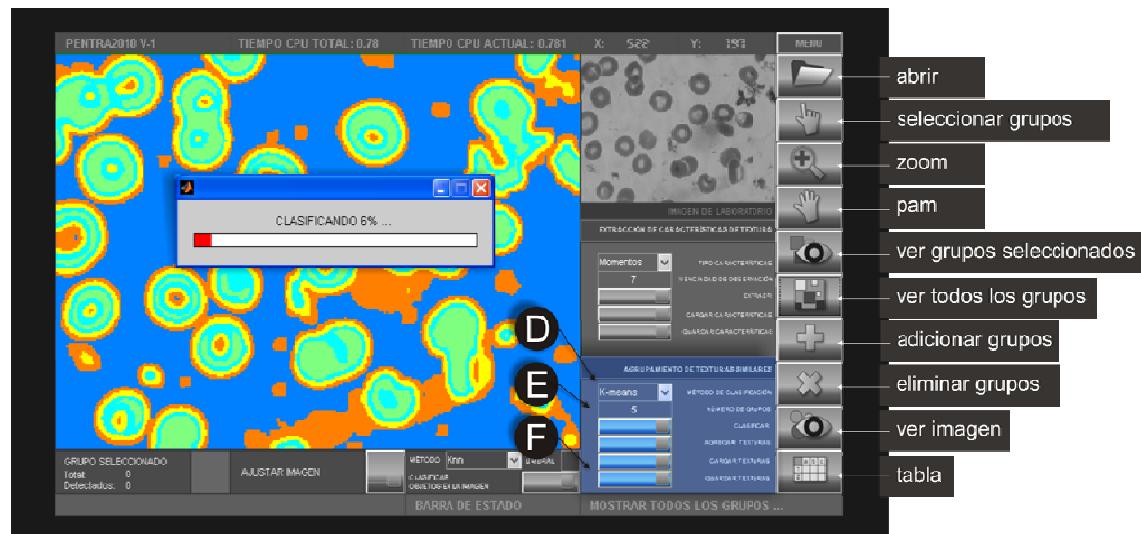


Figura 2.15. Proceso de clasificación.

Para la selección de los grupos pertenecientes a los objetos de interés se creó un menú que permite de forma interactiva realizar este proceso. En este punto el usuario del sistema selecciona a partir del botón de *selección de grupos* del menú los grupos en la imagen principal que desea agregar a la clase de los objetos de interés de la matriz de aprendizaje. Con el botón *eliminar* se permite quitar el grupo seleccionado de la clase de los objetos de interés. Los resultados de este proceso se podrán ir chequeando a través de la tabla disponible en la interfaz como se muestra en la Figura 2.17.

Luego de haber seleccionado los grupos de texturas, el sistema en (F) brinda la oportunidad de incorporar los vectores clasificados a la matriz de aprendizaje, además de guardar y cargar matrices de aprendizaje de un directorio especificado.

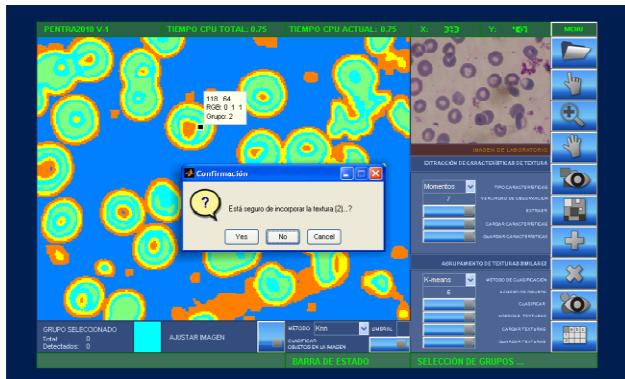


Figura 2.16. Selección de los grupos en la imagen

M	Media	Desviación	Suavizado	Tercer Momento	Uniformidad	Entropía	clases
1	106.5250	2.5469	0.0079	-0.1713	0.0810	4.3955	1
2	141.7900	22.2099	0.0079	-0.1267	0.0509	4.5910	1
3	135.1716	25.4401	0.0099	-0.0010	0.1260	3.2860	1
4	149.1980	3.4207	2.1236e-04	-0.0010	0.1260	3.2860	1
5	158.1700	1.7040	5.0000e-04	-2.9000e-04	0.0260	2.1495	1
6	142.7311	4.1940	3.0698e-04	-0.0014	0.1050	3.5807	1
7	94.8971	5.9589	5.5702e-04	0.0059	0.0876	3.6917	1
8	132.8900	8.4700	0.0011	0.0011	0.0876	3.6917	1
9	132.1980	16.8133	0.0044	-0.0289	0.0445	4.7253	1
10	82.3396	7.0916	9.3076e-04	0.0159	0.0843	3.8977	1
11	109.9032	26.3438	0.0124	0.1541	0.0431	4.8042	1
12	125.8000	24.2000	0.0089	0.0000	0.1260	3.2860	1
13	151.7924	5.7471	3.2628e-04	-0.0009	0.1269	3.2870	1
14	123.8222	9.4788	0.0015	0.0003	0.0874	4.3806	1
15	144.8000	7.8450	1.4854e-04	-2.3020e-04	0.0195	4.1959	1
16	147.5949	18.5303	0.0053	0.1485	0.0791	4.2307	1
17	147.4063	11.4595	0.0021	-0.0491	0.0728	4.1363	1
18	127.9205	26.7135	0.0110	-0.0257	0.0449	4.7226	1
19	139.8000	15.1100	0.0032	0.0000	0.1260	3.2860	1
20	119.7009	20.6729	0.0110	0.0954	0.0447	4.7054	1
21	156.4075	1.8344	6.5156e-05	-1.4913e-04	0.1921	2.6641	1
22	125.8000	24.2000	0.0089	0.0000	0.1260	3.2860	1
23	113.8027	6.4467	7.7077e-04	0.0055	0.0337	3.8955	1
24	153.0860	13.7009	0.0029	-0.0915	0.0688	3.7685	1
25	113.8903	19.2446	0.0059	0.0974	0.0602	4.6265	1
26	109.7000	15.1100	0.0030	0.0000	0.1260	3.2860	1
27	102.4531	7.7428	0.0155	0.0648	3.9968	1	

Figura 2.17. Tabla para la visualización de la matriz de aprendizaje.

2.8.2 Interfaz del proceso de análisis.

El objetivo fundamental de esta interfaz es realizar el conteo de las deformaciones de los eritrocitos presentes en una muestra de manera automática. Para ello se carga una matriz de aprendizaje obtenida en el proceso anterior a partir de una dirección especificada. Posteriormente se realiza el análisis de la imagen (esto incluye detección, segmentación y análisis morfológico de la muestra) a partir de la selección en (B) y en (C) de los métodos de detección y segmentación, respectivamente.

Los resultados del conteo son mostrados en (D), y en (E) se realiza una representación gráfica de los mismos teniendo en cuenta la relación existente entre los coeficientes elíptico y de compacidad descrita en el Capítulo 1 Epígrafe 1.4. Dicha gráfica es mostrada en la Figura 2.19.

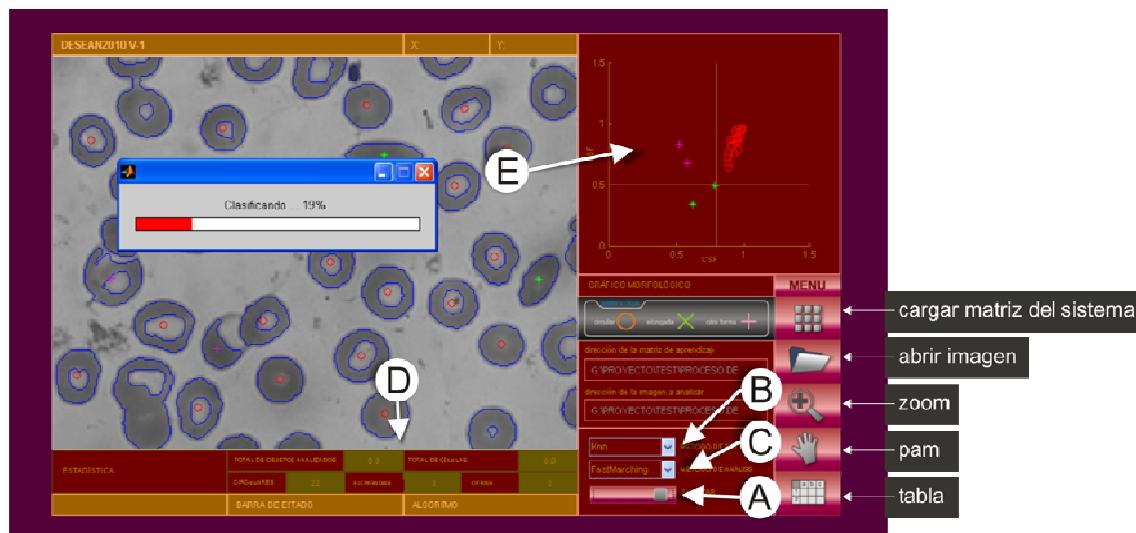


Figura 2.18. Proceso de análisis.

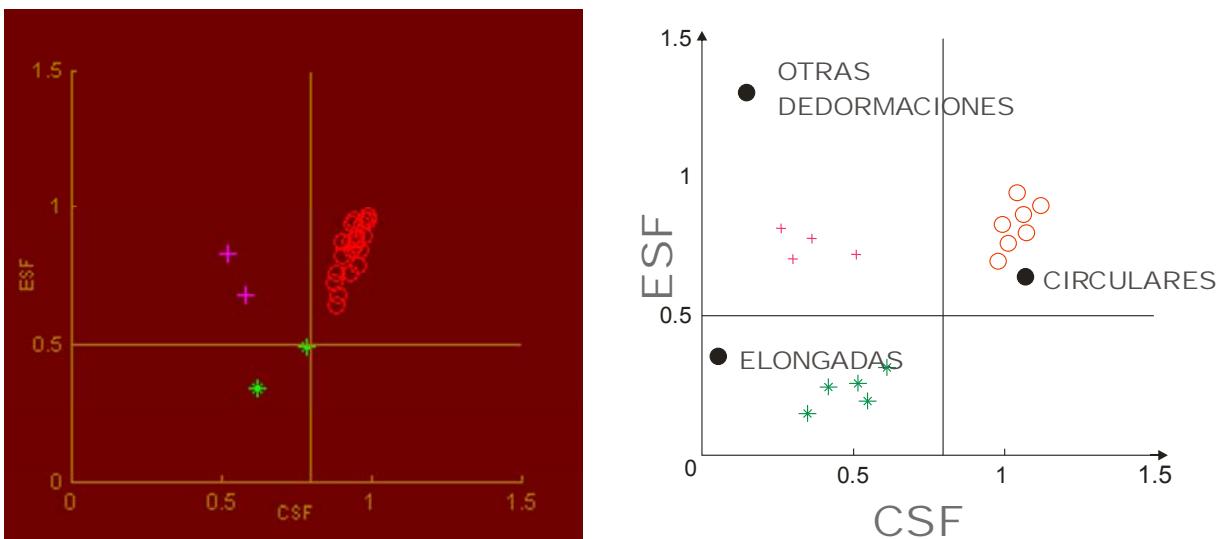


Figura 2.19. Gráfica para la relación ESF-CSF.

2.10 Conclusiones parciales.

En este capítulo se detalla la metodología a seguir para darle cumplimiento a los objetivos planteados, las cuestiones de implementación de los algoritmos propuestos para realizar la extracción de características, es decir, el momento de orden k, la matriz de co-ocurrencia y los modelos de Gabor; así como la de los algoritmos de agrupamiento de texturas tales como K-means y β_0 -conexo y los métodos realizados para la creación, unión y reducción de matrices de aprendizaje. De igual forma para los métodos de detección de texturas pertenecientes a las regiones de interés se detallan los algoritmos k-NN y votación, posteriormente los algoritmos de segmentación Fast Marching y Level Set para obtener los contornos de las regiones detectadas; y un análisis morfológico que permite determinar la cantidad de eritrocitos deformados presentes en una imagen. Además se detallan cuestiones acerca de la implementación del prototipo de software desarrollado en dos interfaces para aplicar los mismos.

CAPÍTULO 3. RESULTADOS Y VALIDACIÓN.

3.1 Introducción.

En este capítulo se hace un análisis de los resultados obtenidos para cada uno de los métodos y estrategias aplicadas en cada fase. Para la fase de entrenamiento se determinaron los métodos de extracción de características adecuados para la investigación, se realizó una comparación entre las estrategias de agrupamiento aplicadas en cuanto a las ventajas y desventajas de cada una y se determinó la más apropiada para emplear en el proceso. En la fase posterior se seleccionaron el algoritmo de detección y el de segmentación adecuados para este problema. Para el análisis de la forma de los objetos se realizó una calibración de los algoritmos de segmentación empleando los coeficientes elípticos y circulares que corroboró el uso de los valores propuestos en [12]. Finalmente el criterio de un especialista permitió evaluar el proceso en su totalidad.

3.2 Condiciones iniciales para el estudio.

Las imágenes empleadas para el estudio fueron adquiridas en el Departamento de Hematología del Laboratorio Clínico del Hospital General “Dr. Juan Bruno Zayas Alfonso” de la ciudad de Santiago de Cuba, donde a partir de muestras de sangre de varios pacientes en estado de crisis de la enfermedad denominada Anemia Drepanocítica se prepararon 17 láminas de sangre periféricas. Se tomaron en total 45 imágenes de 267x200 a una resolución de 180 px/cm a partir de las muestras obtenidas en un microscopio Leika con objetivo de aumento 100x, empleando una cámara Kodak de 8.0 mega píxeles con cañón AF 3X OPTICAL no profesional. Es importante destacar que las imágenes no fueron obtenidas con todo el rigor requerido para este tipo de estudio, por lo cual no se pudo garantizar por ejemplo un nivel de enfoque uniforme para todas, lo que limitó el máximo rendimiento de los algoritmos de segmentación que se emplearon, no obstante los resultados obtenidos fueron satisfactorios.

Para la fase de entrenamiento, en el proceso de aprendizaje de las matrices fueron seleccionadas 15 imágenes, las restantes 30 fueron empleadas indistintamente en el control de la calidad del proceso. Todas las pruebas fueron realizadas en una

computadora con microprocesador Intel Dual Core a 2.5 Ghz con 1Gb de memoria RAM.



Figura 3.1 Microscopio, cámara y láminas de muestras utilizadas.

3.3 Selección del método de extracción de características.

Para la determinación del método de extracción de características de textura que brinde los mejores resultados se tienen en cuenta los siguientes indicadores: el tiempo empleado para realizar la extracción, el número de rasgos o medidas y la calidad del agrupamiento bajo las mismas condiciones, que es evaluada cualitativamente como Excelente (E), Bien (B), Regular (R) y Mal (M). Estos indicadores fueron evaluados para las 15 imágenes del proceso de entrenamiento, arrojando los siguientes resultados:

Tabla 3.1 Los métodos de extracción y los indicadores con los valores obtenidos para cada uno de ellos.

Métodos	Indicadores		
	Tiempo de corrida en segundos	Cantidad de Rasgos	Calidad del agrupamiento
Momento de orden k	110.44	6	E
Modelos de Gabor	703.25	24	R
Matriz de co-ocurrencia	1183.13	6	M
Momento de orden k, modelos de Gabor	762.37	30	B
Momento de orden k, matriz de co-ocurrencia	1251.73	12	B
Matriz de co-ocurrencia, modelos de Gabor	2880.40	30	B
Momento de orden k, modelos de Gabor, matriz de co-ocurrencia	2898.40	36	B

Como se puede apreciar los resultados muestran la eficiencia del método Momento de orden k respecto al resto de los empleados en cuanto a calidad del agrupamiento y tiempo de ejecución, por lo que éste es el seleccionado como el que debe emplearse para el desarrollo del proceso.

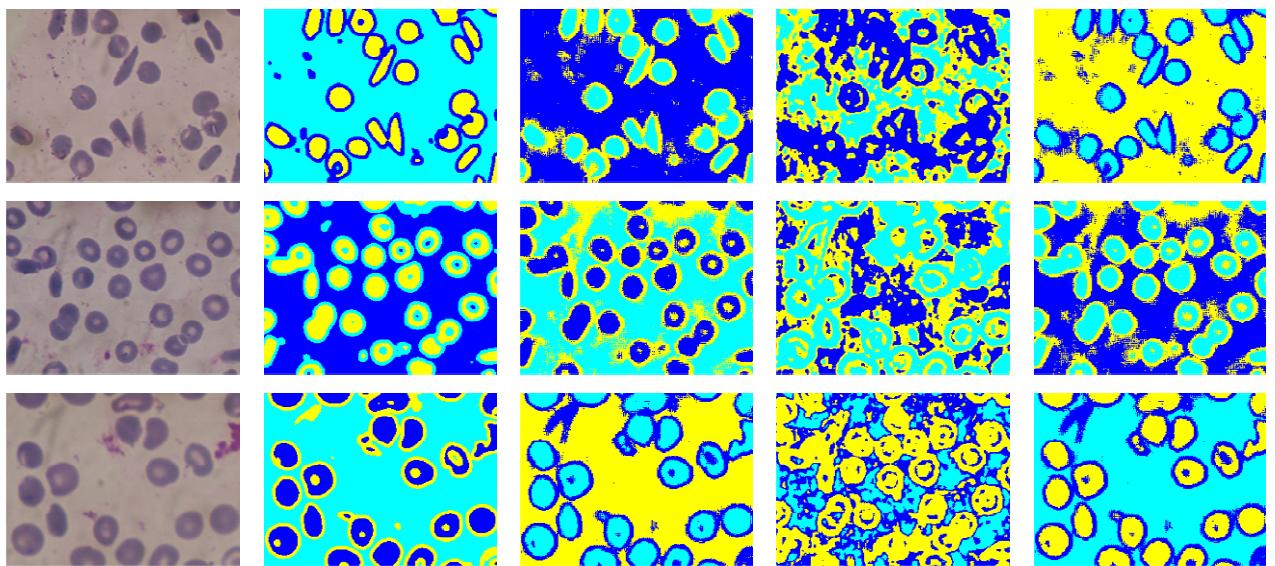


Imagen original. Momento de orden k. Modelos de Gabor Matriz de co-ocurrencia. Todos.

Figura 3.2 Calidad del agrupamiento en 3 imágenes para 4 de los métodos de extracción empleados.

3.4 Valoración de las estrategias de agrupamiento de características.

Posterior a la extracción de los vectores característicos se realiza el agrupamiento de los mismos con el objetivo de obtener un subconjunto representativo de las texturas que aparecen en la imagen que se está analizando. Para esto se emplean dos estrategias que utilizan los algoritmos de agrupamiento K-means y β_0 -conexo. A continuación se detallan las mismas.

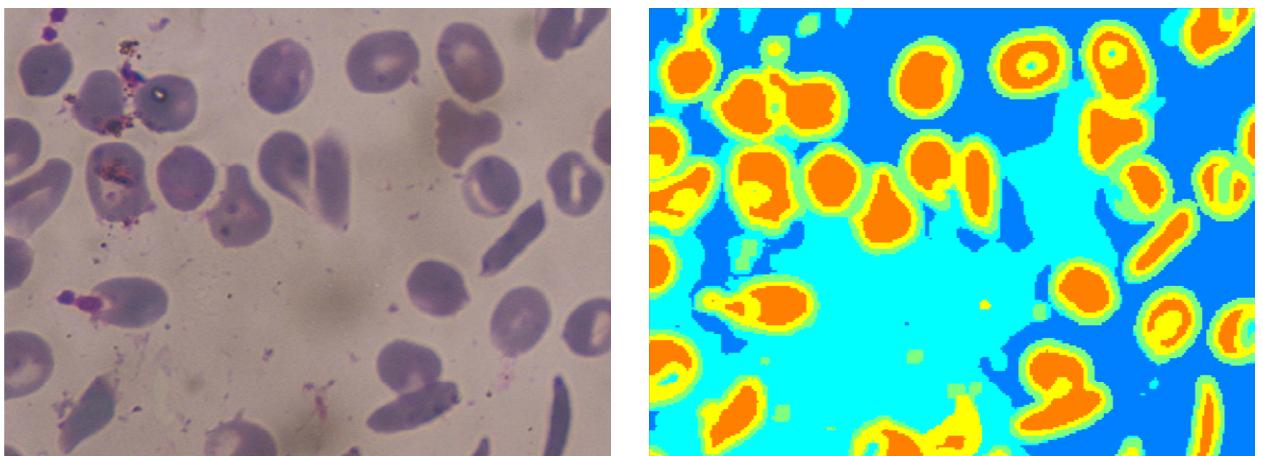
3.4.1 Estrategia de agrupamiento por K-means.

El agrupamiento por K-means en la literatura es uno de los más usados, debido a que es uno de los algoritmos de su tipo con mejor desempeño en cuanto a la velocidad en que crea los grupos y la obtención de los representantes de los mismos. Se realizó un estudio del comportamiento del algoritmo en cuanto a tiempo de corrida para cantidad de grupos $k=3$, $k=10$ y $k=15$ en 14 imágenes de 267x200, lo que representa un total de 53400 vectores por imagen a agrupar. El método con menor tiempo de ejecución fue el momento de orden k, no obstante en todos los casos los resultados fueron razonables, por lo que se determinó procesar toda la imagen. En la Tabla 3.2 se muestran los resultados obtenidos para $k=3$:

Tabla 3.2 Tiempo de corrida en segundos para K-means con $k=3$.

	Momento de orden k	Modelos de Gabor	Matriz de co-ocurrencia	Momento de orden k, modelos de Gabor	Momento de orden k, matriz de co-ocurrencia	Matriz de co-ocurrencia, modelos de Gabor	Momento de orden k., modelos de Gabor, matriz de co-ocurrencia
I1	0.437	1.453	0.680	1.780	0.671	0.870	1.530
I2	0.484	1.109	0.950	1.593	0.765	1.156	1.340
I3	0.546	1.140	0.500	0.750	0.703	1.090	1.078
I4	0.406	1.718	0.620	1.890	0.593	0.937	2.609
I5	0.562	0.687	0.593	0.840	0.515	1.156	0.687
I6	0.578	0.687	0.453	0.687	0.453	1.171	1.078
I7	0.328	0.437	0.531	0.760	0.765	0.828	0.890
I8	0.546	1.000	0.453	1.010	0.64	0.828	1.046
I9	0.562	2.750	0.578	0.718	0.406	0.546	3.671
I10	0.390	0.730	0.578	0.812	0.546	0.937	0.796
I11	0.468	0.510	0.281	0.609	0.568	0.578	0.671
I12	0.453	0.530	0.343	0.703	0.421	0.765	0.828
I13	0.312	0.730	0.640	0.796	0.406	1.125	1.109
I14	0.750	0.900	0.687	0.734	0.890	1.812	1.375
Promedio	0.487	1.027	0.563	0.977	0.595	0.985	1.336

El subconjunto de vectores representativos de las texturas de interés estará formado para este método por los centroides de cada uno de sus grupos. Un ejemplo de los resultados del proceso de agrupamiento por este método para $k=5$ a partir de la extracción de los vectores característicos por momento de orden k para una imagen de dimensiones 267x200 se ilustra en la Tabla 3.3.



a) Imagen Original.

b) Agrupamiento.

Figura 3.3. Agrupamiento de los vectores característicos de una imagen por K-means para $k=5$.

Tabla 3.3. Vectores característicos correspondientes a la Figura 3.3 (b).

	Media	Desviación Estándar	Suavidad	Tercer Momento	Uniformidad	Entropía
T1	156.4269	3.2606	3.1628e-04	-0.0070	0.1774	2.8429
T2	145.1239	3.9960	3.8144e-04	-0.0056	0.1367	3.2131
T3	138.6517	19.8427	0.0064	-0.0930	0.0559	4.4903
T4	118.2907	18.0328	0.0057	0.0615	0.0548	4.5119
T5	100.2249	6.7611	0.0011	0.0215	0.1071	3.6211

El conjunto de vectores obtenidos debe particionarse en representantes de objetos de interés (*OI*) y no representantes de los mismos (*ONI*). En este caso los objetos de interés son los eritrocitos, y para el ejemplo anterior los vectores que representan texturas correspondientes a los mismos son *T4* y *T5*, que deben ser clasificados como *OI* y el resto como *ONI*. Para ello el sistema inicialmente clasificará todos los vectores representantes como *ONI* y permitirá que el usuario seleccione cuáles de ellos clasificar como *OI*. El resultado de este proceso dará lugar a la matriz de aprendizaje para la imagen que se esté analizando. La matriz resultante para el ejemplo anterior se muestra a en la Tabla 3.4.

El sistema permite que el usuario varíe el número de grupos que desea obtener por este algoritmo, con el propósito de que interactivamente se puedan discriminar texturas similares en objetos diferentes. Esta acción en algunos casos no es posible, lo que en cierta medida limita el proceso, aún más cuando los objetos son desde el punto de vista de su textura muy parecidos.

Tabla 3.4. Matriz de aprendizaje obtenida a partir del proceso de agrupamiento por K-means.

	Media	Desviación Estándar	Suavidad	Tercer Momento	Uniformidad	Entropía	Clases
T1	156.4269	3.2606	3.1628e-04	-0.0070	0.1774	2.8429	ONI
T2	145.1239	3.9960	3.8144e-04	-0.0056	0.1367	3.2131	ONI
T3	138.6517	19.8427	0.0064	-0.0930	0.0559	4.4903	ONI
T4	118.2907	18.0328	0.0057	0.0615	0.0548	4.5119	OI
T5	100.2249	6.7611	0.0011	0.0215	0.1071	3.6211	OI

Para la investigación realizada se emplearon las 15 imágenes de la fase de entrenamiento, de forma experimental se determinó la cantidad de grupos como $k=15$. Se obtuvieron 15 matrices de aprendizaje, que fueron unidas y reducidas mediante el proceso descrito en el Capítulo 1 epígrafe 1.2.3, obteniéndose la matriz que representa a las texturas de interés para la estrategia por K-means, mostrada en la Tabla 3.5.

Tabla 3.5. Matriz de aprendizaje obtenida por el proceso de agrupamiento K-means.

	Media	Desviación Estándar	Suavidad	Tercer Momento	Uniformidad	Entropía	clase
T1	133.6135	23.5093	0.0086	-0.0696	0.0479	4.6609	ONI
T2	115.0449	20.1538	0.0064	0.0711	0.0405	4.8493	ONI
T3	95.2471	14.8549	0.0035	0.0510	0.0515	4.5566	ONI
T4	113.2565	4.1681	3.4070e-04	0.0032	0.1227	3.3492	ONI
T5	142.6938	4.0499	3.1836e-04	-0.0032	0.1262	3.3235	ONI
T6	125.1736	14.6941	0.0035	0.0116	0.0490	4.6035	ONI
T7	154.7382	2.0969	8.7171e-05	-3.8219e-04	0.2047	2.6258	ONI
T8	97.8181	5.2508	5.1516e-04	0.0070	0.1057	3.5899	OI

3.4.2 Estrategia de agrupamiento por β_0 -conexo.

La aplicación del K-means tiene como principal desventaja que debe fijarse la cantidad de grupos a obtener, por este motivo se determinó emplear la estrategia basada en el método β_0 -conexo, que además logra discriminar mejor que el K-means las texturas correspondientes a cada grupo.

Este método no se aplicó a toda la imagen, dado que el tiempo promedio observado para el mismo en 10 imágenes de dimensiones 267x200 fue de 498.0 segundos, por lo que sería prácticamente intratable aplicarlo a la imagen completa, luego lo que se propone es seleccionar pequeñas regiones de la misma donde confluyan las texturas representativas, tanto de objetos de interés como de otros existentes, para realizar el proceso. Se realiza el agrupamiento a cada región de la imagen seleccionada y se crea la matriz de aprendizaje para cada una de ellas, las que posteriormente serán unidas y reducidas para crear la que representa a las texturas de interés para la imagen. Un ejemplo de una es la obtenida a partir del agrupamiento en las regiones seleccionadas dentro de la imagen (ver Figura 3.4) por esta estrategia se muestra en la Tabla 3.6.

Las matrices de aprendizaje obtenidas por este método van a caracterizar más rigurosamente los grupos, dada la cantidad y representatividad de los vectores que la conforman, lo que es deseable en este caso, aunque esta cantidad podría ser en ocasiones excesivamente grande en dependencia de los datos de entrada (dígase tamaño de la región analizada, valor del β_0 seleccionado, etc.).

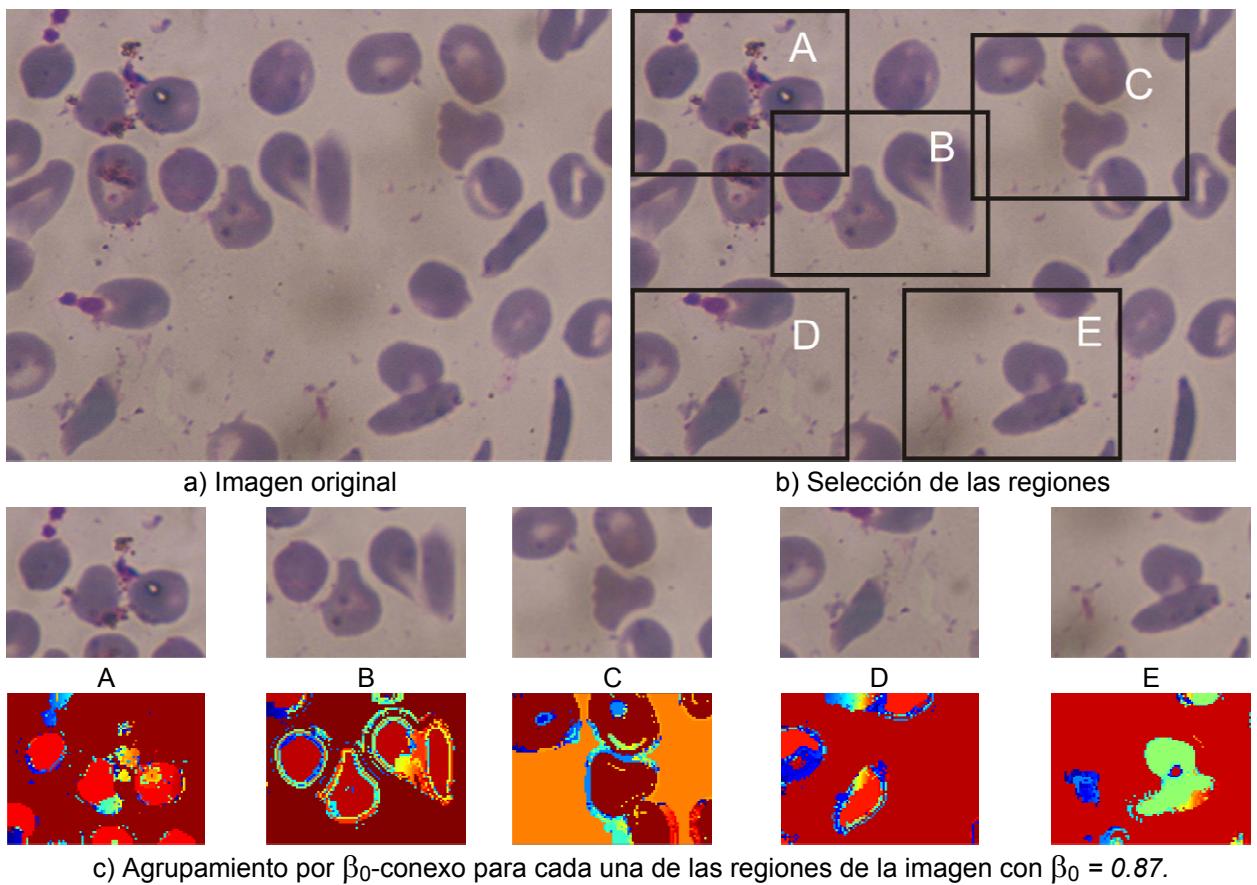


Figura 3.4 Selección y agrupamiento de regiones de la imagen.

Tabla 3.6. Matriz de aprendizaje para las 5 regiones de interés de la Figura 3.4.

	Media	Desviación Estándar	Suavidad	Tercer Momento	Uniformidad	Entropía	clase
T1	123.9184	22.1478	0.0075	0.0797	0.0471	4.6377	ONI
T2	132.7823	17.5811	0.0047	-0.0546	0.0668	4.3015	ONI
T3	116.5306	11.4326	0.0020	0.0150	0.0579	4.3697	ONI
T4	118.0408	14.5293	0.0032	0.0415	0.0408	4.8107	ONI
T5	124.7551	14.3826	0.0032	0.0104	0.0440	4.7341	ONI
T6	121.2245	18.9487	0.0055	0.0244	0.0435	4.7300	ONI
T7	108.9082	15.4384	0.0037	0.0641	0.0525	4.6193	ONI
T8	91.0816	14.4432	0.0032	0.0777	0.0537	4.4919	ONI
T9	120.2449	19.8695	0.0060	0.0167	0.0354	4.9562	ONI
T10	137.5918	14.3838	0.0032	-0.0350	0.0471	4.6377	ONI
T11	130.0714	18.4607	0.0052	-0.0670	0.0621	4.3085	ONI
T12	147.2245	5.1263	4.0432e-04	-0.0024	0.0887	3.7805	ONI
T13	117.6327	17.0665	0.0045	0.0220	0.0392	4.8736	ONI
T14	115.5646	14.1013	0.0030	0.0248	0.0423	4.7887	ONI
T15	101.1633	19.7519	0.0060	0.0708	0.0437	4.7113	ONI
T16	96.1020	4.6390	3.3084e-04	0.0012	0.0787	3.8439	OI
T17	93.1866	2.2297	7.8266e-05	2.6113e-04	0.1742	2.8492	OI
T18	99.4490	4.9941	3.8342e-04	1.2725e-04	0.0629	4.1325	OI
T19	96.7347	2.1557	7.2425e-05	6.9149e-05	0.1458	2.9658	OI

Para la investigación realizada se emplearon las 15 imágenes de la fase de entrenamiento, de forma experimental se determinó el valor del $\beta_0 = 0.87$. Se obtuvieron 15 matrices de aprendizaje, que fueron unidas y reducidas obteniéndose la matriz que

representa a las texturas de interés para la estrategia por β_0 -conexo, mostrada en la Tabla 3.7.

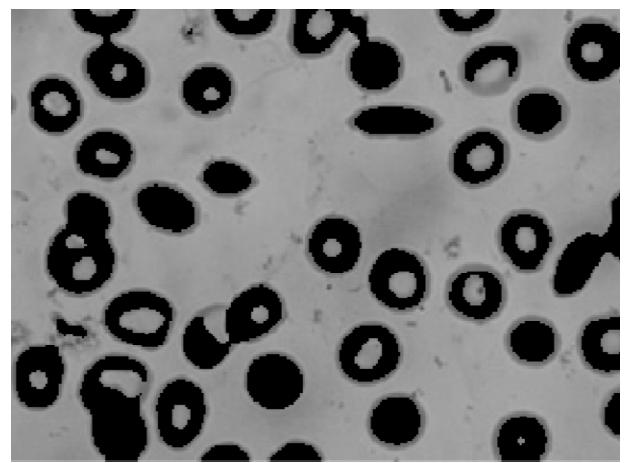
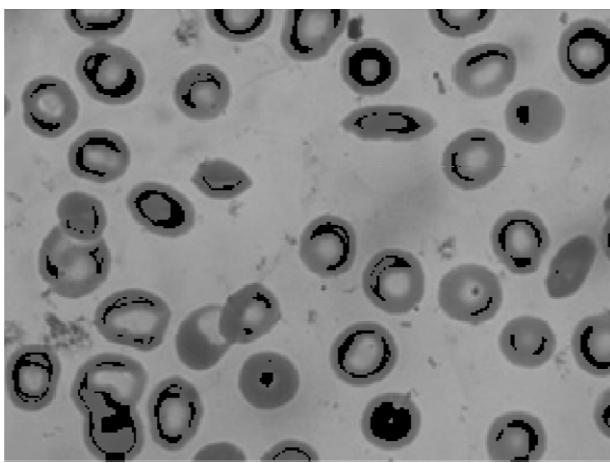
Tabla 3.7 Matriz de aprendizaje obtenida por el proceso de agrupamiento β_0 -conexo.

	Media	Desviación Estándar	Suavidad	Tercer Momento	Uniformidad	Entropía	clase
T1	105.7398	11.9461	0.0022	-0.0030	0.0473	4.5890	ONI
T2	156.6769	7.6690	9.0441e-04	-0.0216	0.1383	3.2309	ONI
T3	104.3776	12.6771	0.0025	0.0614	0.0707	4.1406	ONI
T4	113.4694	26.9536	0.0110	0.2551	0.0471	4.5989	ONI
T5	150.3946	11.2967	0.0020	-0.0584	0.1162	3.4939	ONI
T6	126.0816	30.4578	0.0141	0.0020	0.0387	4.8619	ONI
T7	134.8878	29.3986	0.0131	-0.1208	0.0508	4.5852	ONI
T8	147.9796	22.4833	0.0077	-0.2300	0.0704	4.1244	ONI
T9	97.8513	5.4175	4.5212e-04	0.0025	0.0817	3.8801	OI
T10	104.3061	8.3214	0.0011	0.0224	0.0968	3.6786	OI
T11	72.4898	8.6164	0.0011	0.0185	0.0941	3.7681	OI

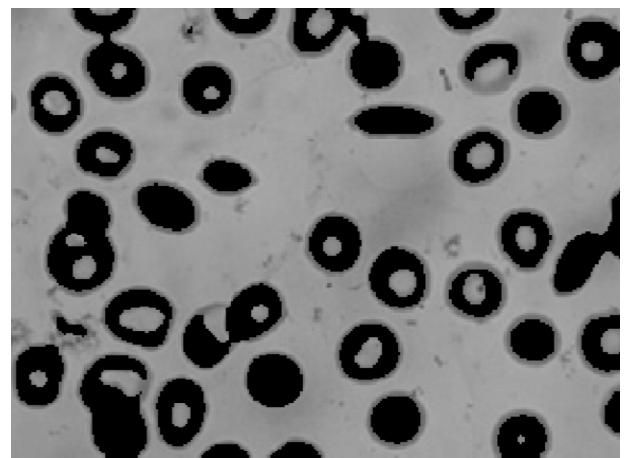
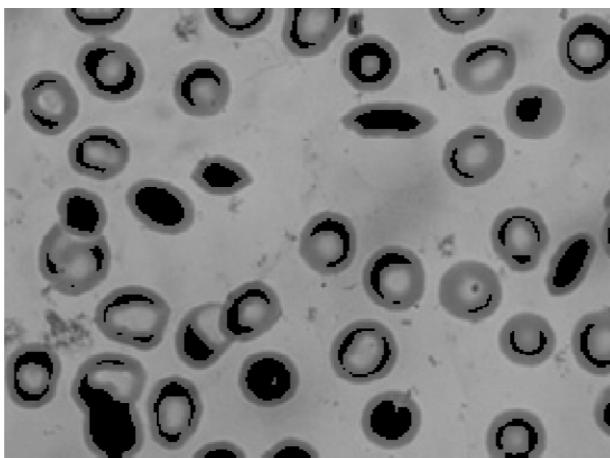
3.5 Detección inicial de objetos de interés.

En este punto se desarrolló un proceso de pruebas para validar la efectividad de la detección inicial de texturas pertenecientes a objetos de interés en la imagen, empleando las dos matrices de aprendizaje obtenidas por los métodos de agrupamiento utilizando K-means y β_0 -conexo, considerando objetos de interés en la imagen a las células o eritrocitos. Para la detección se utilizaron los algoritmos k-NN y votación descritos en el Capítulo 1 epígrafe 1.2.4. Una muestra de los resultados obtenidos se ilustra en la Figura 3.5.

Las pruebas de detección se realizaron en las 30 imágenes destinadas al control de la calidad del proceso en general. En este estudio se contó con el criterio de la especialista en primer grado en MGI y Laboratorio Clínico, Dra. Karelis Fernández Arias, responsable del Departamento de Hematología Especial del Hospital General Dr. Juan Bruno Zayas Alfonso, la cual realizó manualmente el análisis de 472 células presentes en 30 muestras, resultado que será utilizado en la investigación para determinar la calidad del proceso. Se consideró en este punto para el total de células existentes, cuántas fueron detectadas y cuántas no, además de los objetos que no eran células y también fueron detectados.



a) Detección por k-NN y votación a partir de la estrategia K-means.

b) Detección por k-NN y votación a partir de la estrategia β_0 -conexo.**Figura 3.5** Detección de regiones de interés a partir de las matrices obtenidas.

A partir de estos resultados la calidad de la detección fue estimada por la aplicación del siguiente funcional de calidad:

$$\phi(P) = \frac{OBC}{OBC + OMC + OAC} \quad (3.1)$$

donde OBC es la cantidad de células detectadas, OAC es la cantidad de células no detectadas y OMC es la cantidad de objetos detectados que no son células. Los resultados obtenidos por el mismo se ilustran en la Tabla 3.8.

Tabla 3.8. Resultados de la detección de objetos de interés por k-NN.

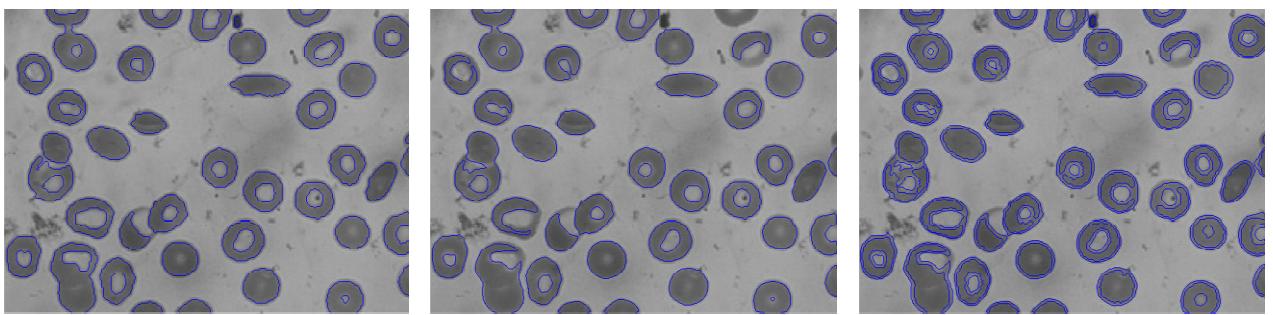
Método		OBC	OMC	OAC	TOTAL (OBC+OAC)	%
k-NN	K-means	454	16	18	472	93.0
	β_0 -conexo	460	17	12	472	94.0
VOTACIÓN	K-means	472	66	0	472	87.7
	β_0 -conexo	472	75	0	472	86.2

Como se puede observar los resultados del algoritmo de votación no fueron los esperados, aunque logró detectar un 100% de los objetos de interés no fue lo suficientemente estricto como para discriminar objetos no deseados en la imagen. Esto se debe a que para el empleo del algoritmo se determinaron como conjuntos de apoyo los métodos de extracción de características empleados, pero en el epígrafe 3.3 se estableció como único método a emplear el momento de orden k , con lo cual se tiene un solo conjunto de apoyo y según [7] esto provoca que el algoritmo de votación no brinde sus mejores resultados, debido a la naturaleza del mismo.

De los resultados mostrados en la Tabla 3.8 se determinó que el empleo de la matriz de aprendizaje obtenida por la estrategia del β_0 -conexo y la aplicación del algoritmo k-NN para la detección de regiones iniciales fueron los métodos más efectivos a emplear para el desarrollo de esta investigación. No obstante se realizaron las pruebas para las dos matrices de aprendizaje obtenidas, por los algoritmos de K-means y β_0 -conexo.

3.6 Resultados de la segmentación de los objetos en la imagen.

Posterior a la detección efectuada en el proceso anterior se realiza la segmentación de las 30 imágenes destinadas al control de la calidad por los métodos Fast Marching, Level Set y la combinación de los mismos que fueron descritos en el Capítulo 1 epígrafe 1.3.4. En este trabajo se realizó un estudio de estos algoritmos en el que se valoró la calidad y el tiempo de la segmentación realizada para las 30 imágenes analizadas por la especialista. Estos resultados no son concluyentes pero dan la medida de cuán convenientes podrían ser para este estudio, de cualquier forma los tres métodos serán empleados para el análisis de la calidad de todo el proceso. Los resultados se muestran en la Figura 3.6.



(a) Fast Marching.

(b) Level Set.

(c) Fast Marching, Level Set.

Figura 3.6 Contornos obtenidos por cada uno de los métodos de segmentación empleados.**Tabla 3.9.** Tiempo de corrida para la segmentación mostrada en la Figura 3.4.

Métodos	Tiempo en segundo para imágenes de 267x200
Fast Marching	11.4531
Level Set	1018.7
Fast Marching, Level Set	829.5313

Tabla 3.10. Umbrales determinados para cada método de segmentación.

Métodos	Umbrales		
Fast Marching	0.5		
Level Set	0.005		
Fast Marching, Level Set	1.0	0.005	

Este estudio demostró que efectivamente el algoritmo Fast Marching es el más rápido, como se muestra en el ejemplo de la Tabla 3.9, pero los contornos obtenidos por el mismo son menos suaves que los obtenidos por el Level Set y que de la combinación de ambos se obtiene un método de segmentación que une las cualidades de estos.

Estos algoritmos dependen de la estimación de valores umbrales para la velocidad del desplazamiento de la curva en los bordes de los objetos. Los valores de los umbrales fueron seleccionados de forma experimental y se muestran en la Tabla 3.10.

3.7 Calibración de los coeficientes morfológicos para la clasificación de formas en los eritrocitos de las muestras de sangre.

A partir de los contornos obtenidos en la segmentación realizada por los métodos propuestos, se hace un análisis morfológico de los mismos en aras de contar cada una de las formas presentes en los eritrocitos que son de interés para este estudio, las mismas son: de forma “Circular” correspondiente a eritrocitos normales y “Elongadas” o en forma de hoz correspondiente a los drepanocitos (una de las deformaciones de los eritrocitos), todos los eritrocitos que presenten cualquier otra forma serán contados

como pertenecientes a otras deformaciones (ahí se incluyen los que presentan forma de estrella o equinocitos).

Se pueden definir entonces tres clases a las cuales pertenecerán cada uno de los contornos obtenidos producto de la segmentación: Circulares, Elongadas y Otras Deformaciones. La pertenencia de cada contorno a una de estas clases está determinada por el valor de los coeficientes de compacidad (CSF) y elíptico (ESF) calculados para cada uno de ellos y que fueron descritos en el Capítulo 1 epígrafe 1.4. Los valores determinados para estos fueron obtenidos en [12], donde se hace un análisis de las deformaciones para este tipo de células.

Un aspecto importante en este punto fue determinar si los valores para estos coeficientes son factibles para la segmentación realizada por los algoritmos que aquí se proponen. Para ello se realizó una calibración de los mismos a partir de imágenes creadas manualmente, a las cuales se clasificó a través del sistema y se analizaron los resultados del mismo. La siguiente figura muestra los resultados de la calibración.

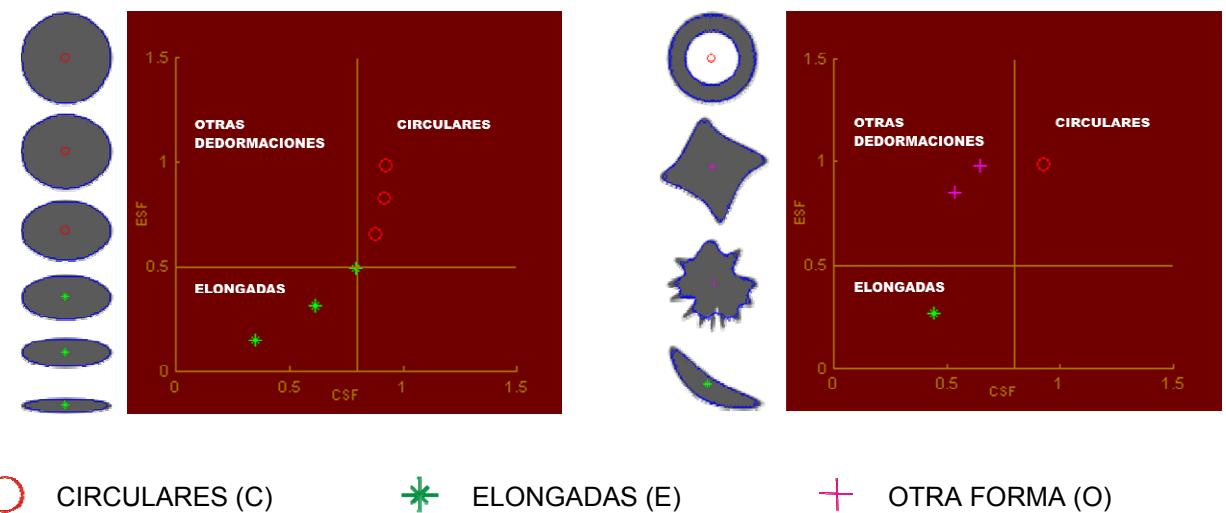


Figura 3.7. Resultados de la calibración para formas artificiales.

Para el análisis de cada imagen se consideró tomar como objetos válidos aquellas regiones segmentadas cuya área fuese mayor que la mitad del área promedio de todos los objetos segmentados en la imagen y menor que el doble de la misma. Las regiones en las que el área fuera más pequeña o más grande que el promedio se corresponden

con ruidos u otros artefactos, así como con objetos solapados que puedan quedar segmentados sin separarse, por tanto los análisis morfológicos no son válidos.

Con esta estrategia se logró eliminar el 100% de las regiones detectadas por el algoritmo k-NN que no pertenecían a las zonas de interés para las 30 imágenes evaluadas por la especialista. Este proceso permitió que aumentara la efectividad de la detección del método, por ejemplo en el caso de la detección por k-NN con β_0 -conexo la efectividad aumentó de 94.0% a un 97.45% después de haber eliminado las 17 regiones que no pertenecían a zonas de interés.

3.8 Conteo de eritrocitos deformados. Validación de resultados.

Para cuantificar el rendimiento de un método de clasificación, es necesario llevar a cabo experimentos de validación. La validación de un algoritmo de clasificación consiste en determinar la calidad de la clasificación del mismo. Para realizar tal proceso existen varias técnicas: con muestras de control, por el criterio de un especialista, utilizando estadígrafos, entre otras [13].

En este caso se hizo uso del criterio de la especialista, que realizó la clasificación manualmente de las formas de cada una de las 472 células presentes en las 30 imágenes evaluadas. La Figura 3.8 muestra una imagen clasificada por la especialista y el resultado de la clasificación por el sistema sobre la misma imagen para los tres métodos de segmentación utilizados. Se empleó el funcional de calidad descrito por la ecuación (3.1) para evaluar estos resultados.

Para medir la calidad del proceso de clasificación se analizaron los resultados del mismo teniendo en cuenta sólo los objetos de interés o eritrocitos detectados inicialmente, no así la totalidad de las células, que incluye las que fueron determinadas por la especialista como células y que no fueron detectadas por el sistema. En las Tabla 3.11 y Tabla 3.12 se muestran los resultados para cada uno de los métodos de segmentación empleados, donde CODI y TCA significan cantidad de objetos detectados inicialmente y total de contornos obtenidos para el análisis respectivamente.

En el epígrafe 3.5 se determinó que los mejores resultados en la detección se obtuvieron para la matriz de aprendizaje del método β_0 -conexo, por lo que es de

esperar mejores resultados en la clasificación a partir de la detección con esta matriz. Sin embargo los resultados mostrados en las Tabla 3.11 y Tabla 3.12 no coinciden en su totalidad con este supuesto. Esto se debe al método empleado para el análisis morfológico y las características particulares de los algoritmos de segmentación utilizados.

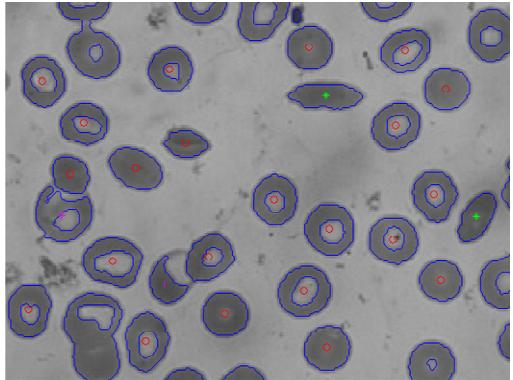
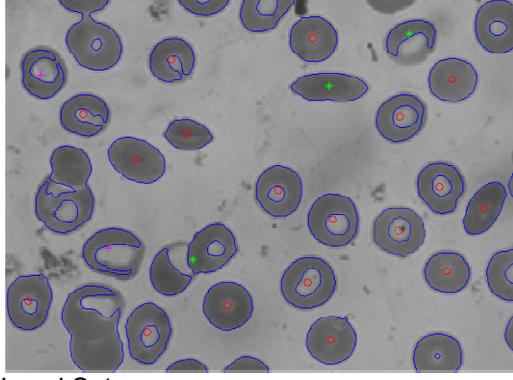
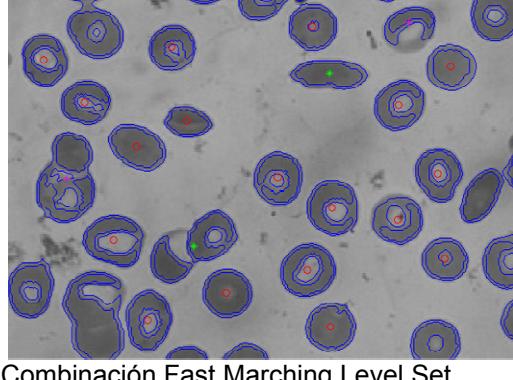
Clasificación manual	Clasificación automática	Resultados
		Cantidad de células circulares (C/O): 23/21. Cantidad de células elongadas (E/*): 3/2. Cantidad de células con otras deformaciones (O/+): 1/1.
		Cantidad de células circulares (C/O): 23/16. Cantidad de células elongadas (E/*): 3/1. Cantidad de células con otras deformaciones (O/+): 1/0.
		Cantidad de células circulares (C/O): 23/17. Cantidad de células elongadas (E/*): 3/1. Cantidad de células con otras deformaciones (O/+): 1/1.
Combinación Fast Marching Level Set		

Figura 3.8. Clasificación de los eritrocitos en una muestra de sangre.

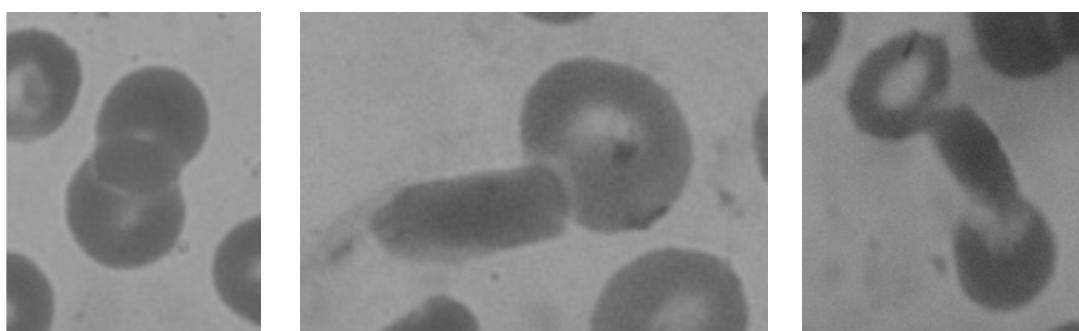
Tabla 3.11. Resultados del análisis para la estrategia K-means.

	OBC	OMC	OAC	CODI	TCA	%
Fast Marching	334	78	15	454	427	78.22
Level Set	298	83	10	454	391	76.21
Fast Marching, Level Set	312	67	37	454	416	75.00

Tabla 3.12. Resultados del análisis para la estrategia β_0 -conexo.

	OBC	OMC	OAC	CODI	TCA	%
Fast Marching	354	72	15	460	441	80.27
Level Set	280	113	21	460	414	66.82
Fast Marching, Level Set	288	95	43	460	426	67.60

Los métodos de segmentación empleados no son capaces de solucionar los casos de solapamiento entre objetos, que en estas muestras de sangre es frecuente observar debido a la forma de preparación de las mismas (ver Figura 3.9). En el caso del Fast Marching la condición de parada es más exigente que los otros métodos, por lo que en algunas ocasiones logra separar los contornos correspondientes a células solapadas, aunque los bordes detectados son menos precisos que en los otros casos. Sin embargo, los métodos Level Set y la combinación de Level Set con Fast Marching logran obtener bordes muy precisos, pero en los casos de células solapadas nunca logran separarlas, por lo que aportan mayor cantidad de casos de solapamiento (ver Figura 3.10). Como el análisis morfológico utilizado no es sensible al solapamiento, al no segmentarse de forma correcta la célula el sistema realiza una mala clasificación de la misma, o en el mejor de los casos se abstiene.

**Figura 3.9.** Casos de solapamiento en las muestras obtenidas.

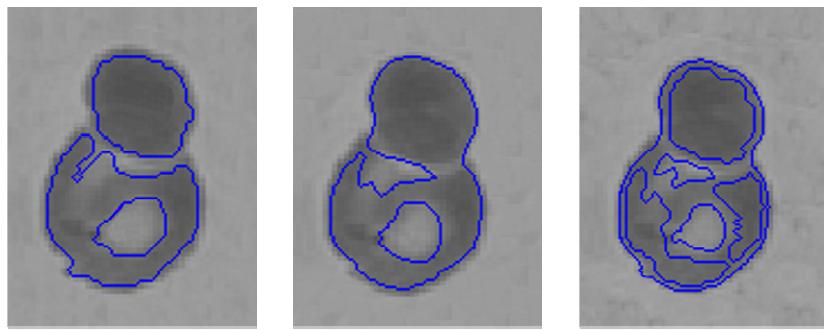


Figura 3.10. Segmentación de células solapadas por Fast Marching, Level Set y la combinación de ambos respectivamente.

En el caso de β_0 -conexo con el algoritmo de segmentación de Fast Marching se obtuvo un 80.27% de calidad en el proceso de clasificación, sin embargo para los otros métodos de segmentación los resultados con esta estrategia estuvieron por debajo de los obtenidos con la estrategia de K-means, lo que contradice el supuesto de que los mejores resultados se deben obtener con la estrategia de β_0 -conexo, pero se corresponde con lo explicado acerca de la capacidad de estos dos métodos de segmentación para separar células solapadas, al no ser capaces de manejar estos casos el error en la clasificación de células por parte del sistema es mayor. De esto se desprende que para este caso es conveniente emplear el método de segmentación Fast Marching a pesar de que sus resultados son menos precisos, teniendo en cuenta que el análisis morfológico realizado es sencillo.

La calidad del proceso de clasificación se analizó también respecto a la totalidad de las células existentes en las imágenes valoradas por la especialista, empleando la clasificación manual que la misma realizó de 472 células presentes en las 30 imágenes evaluadas. Los resultados se muestran en las siguientes tablas donde OAS son las abstenciones producto de la segmentación y OND son los objetos no detectados.

Tabla 3.13. Resultados del proceso para la estrategia K-means.

	OBC	OMC	OAC		TOTAL	%
			OAS	OND		
Fast Marching	334	78	15	45	472	70.76
Level Set	299	83	10	80	472	63.34
Fast Marching, Level Set	312	67	37	56	472	66.10

Tabla 3.14. Resultados del proceso para la estrategia β_0 -conexo.

	OBC	OMC	OAC		TOTAL	%
			OAS	OND		
Fast Marching	354	72	15	31	472	75.0
Level Set	280	113	21	58	472	59.32
Fast Marching, Level Set	288	95	43	46	472	61.01

Estos resultados se corresponden con los obtenidos para el comportamiento del sistema respecto a los objetos de interés detectados inicialmente (ver Tabla 3.11 y Tabla 3.12), y confirman que la estrategia empleando β_0 -conexo con el algoritmo de segmentación Fast Marching arrojaron los mejores resultados para esta investigación, con un 75% de exactitud en la clasificación.

3.9 Conclusiones parciales.

En este capítulo se mostraron los resultados obtenidos con la aplicación de la metodología propuesta para el conteo de eritrocitos en imágenes de muestras microscópicas de sangre. Se realizó una comparación entre los métodos de extracción de características de texturas que permitió determinar que el momento de orden k es el método que para este problema debe ser utilizado, debido a que con el mismo se obtienen los mejores grupos en un menor tiempo. La comparación entre los resultados a partir del funcional de calidad empleado permitió determinar que el algoritmo k-NN para la matriz de aprendizaje adquirida mediante el β_0 -conexo es el mejor para la detección de objetos de interés dentro de la imagen. Los resultados de la segmentación demostraron la superioridad del Fast Marching en este caso y el análisis de una especialista permitió validar la calidad del proceso, obteniéndose un 75% de fiabilidad, lo que se considera un buen resultado, teniendo en cuenta que el proceso de conteo de células de forma manual es extremadamente engorroso y que las condiciones en que fueron efectuados los experimentos no fueron las mejores, sobre todo en cuanto a captura de imágenes y determinación de parámetros de los algoritmos de forma experimental.

CONCLUSIONES.

Con el desarrollo de este trabajo se arribó a las siguientes conclusiones:

- Se logró determinar el método de extracción de características adecuado a aplicar en el proceso, que resultó ser el momento de orden k de un conjunto de siete valorados.
- Se propuso un método para la reducción de varias matrices de aprendizaje pertenecientes a varias imágenes, que permitió contar con una única matriz para el proceso de detección de texturas en varias imágenes.
- Se determinaron los métodos de clasificación a emplear a partir de la matriz de aprendizaje obtenida, que fueron k-NN y votación.
- Se determinaron los métodos de segmentación de contornos a emplear, Fast Marching y Level set.
- Se determinaron las medidas a emplear para obtener la deformación celular, coeficiente elíptico y coeficiente de compacidad.
- Se implementó un prototipo de software que emplea los métodos definidos para obtener el número de células deformadas dentro de una muestra microscópica de sangre.
- Se validaron estadísticamente los resultados obtenidos, lo que mostró una efectividad en la detección de eritrocitos deformados de un 75% en todo el proceso.

Por tanto se logró cumplir el objetivo general planteado, referido a determinar de forma automatizada la cantidad de eritrocitos deformados en una muestra de sangre, obteniéndose resultados considerados como buenos teniendo en cuenta las condiciones de obtención de las imágenes procesadas y el desarrollo del proceso en general.

RECOMENDACIONES.

A partir del trabajo se proponen las siguientes recomendaciones:

1. Realizar pruebas con métodos que tengan en cuenta el color en el proceso de extracción de características, debido a que los colorantes empleados en la preparación de las muestras permiten diferenciar cada componente de la sangre.
2. Comprobar la efectividad de la matriz de aprendizaje aplicando una matriz de control de calidad construida manualmente, donde se aporten valores pertenecientes a las clases de interés a clasificar.
3. Valorar la creación de conjuntos de apoyo para el algoritmo de votación a partir de las características de las medidas de textura extraídas, aplicando algoritmos de selección de variables como pueden ser los testores típicos, para determinar qué características del momento de orden k pueden considerarse como conjuntos de apoyo y no tomarlas todas como un solo conjunto, lo que afecta la eficiencia del empleo del método en estos momentos.
4. Realizar un estudio para este tipo de imágenes de los distintos valores empleados como umbrales para los métodos de segmentación, así como los umbrales de semejanza empleados y el umbral de tipicidad utilizado en el proceso de reducción de matrices.
5. Valorar la efectividad de realizar un pre-procesamiento de la imagen empleando un algoritmo de suavizado para la eliminación del ruido que permita realizar los bordes de la imagen. Los autores del método proponen el Min-Max en [9].
6. Valorar la implementación del método Narrow Band para el esquema Level Set como variante de optimización del cálculo en este esquema.
7. Valorar la posibilidad de emplear métodos de segmentación que logren el tratamiento de los casos de solapamiento entre células en la imagen.
8. Realizar un estudio de los rangos propuestos en [12] para los coeficientes elíptico y de compacidad en el análisis de forma y así determinar la efectividad de los mismos para este tipo de objeto. Valorar otros métodos que puedan ser más efectivos para análisis morfológico.

REFERENCIAS BIBLIOGRÁFICAS.

- [1] Ramírez Manzanares, Alonso. "Reporte de búsqueda, detección y conteo de objetos". Centro de Investigación en Matemáticas A.C., Guanajuato, México, pp 2-21, 28-33. 2002.
- [2] Shapiro Linda, Stockman George. "Computer Vision". Department of Computer Science, University of Washington. Capítulo 7 pp 236- 247. Marzo 2000.
- [3] Sucar, Luis Enrique. "Visión Computacional". Departamento de Computación ITESM Cuernavaca. Vol 5.
- [4] González, Rafael C. "Digital Image Processing". Second Edition. Electrical Engineering Department. University of Tennessee. Vol 11, pp 463-470, Appendix C, pp 588-591.
- [5] Coto, Ernesto. "Métodos de Segmentación de Imágenes Médicas". Facultad de Ciencias, Escuela de Computación, Universidad Central de Venezuela, Caracas, pp 9-15. Marzo 2003.
- [6] Movellan, Javier R. "Tutorial on Gabor Filters". Machine Perception Laboratory. Septiembre 2008.
- [7] Ruiz Shulcloper, José y Lazo Cortés, Manuel. "Introducción al Reconocimiento de Patrones: enfoque lógico combinatorio". Serie Verde No. 51, CINVESTAV-IPN. México, 1995.
- [8] Kass, M. et al. "Snakes: Active contours models". International Journal of Computer Vision. Vol. 1. No 4. pp. 321-331. 1987.
- [9] Osher, S. and Sethian, J. A. "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi Formulation" . J. Comput. Phys. 79. pp. 12-49. 1988 .
- [10] Heredia, R. "Segmentación de Imágenes Médicas Aplicando Level Sets". Tesis presentada en opción al título de Licenciado en Ciencias de la Computación. Facultad de Matemática y Computación. Universidad de Oriente. 2008.
- [11] Malladi, R., and Sethian, J.A., An O(NlogN) "Algorithm for Shape Modeling, with R. Malladi", Proceedings of the National Academy of Sciences, Vol. 93, pp. 9389-9392, September 1996.

REFERENCIAS BIBLIOGRÁFICAS.

- [12] Toshio Asakura, Aisha T. Nelson. "Percentage of Reversibly and Irreversibly Sickled Cells Are Altered by the Method of Blood Drawing and Storage Conditions". 12 de Diciembre, 1996.
- [13] J. Ruiz S., E. Alba C., M. Lazo C. "Introducción al Reconocimiento de Patrones". Departamento de Ingeniería Eléctrica, Sección de Computación. Centro de Investigaciones y de Estudios Avanzados. Instituto Politécnico Nacional de México. Noviembre 23 de 1994.
- [14] Calzadilla, F. "Segmentación de Imágenes Médicas Aplicando Texturas". Tesis presentada en opción al título de Licenciado en Ciencias de la Computación. Facultad de Matemática y Computación. Universidad de Oriente. 2009.

OTRAS BIBLIOGRAFÍAS.

1. Adalsteinsson, D. and Sethian J. "A Fast Level Set Method for Propagating Interfaces." Lawrence Berkeley Laboratory and Department of Mathematics. University of California. Berkeley, CA 94720. Vol. 118(2), pp. 269-277, September 1994.
2. Adalsteinsson, D. and Sethian J. "The Fast Construction of Extension Velocities in Level Set Methods". Lawrence Berkeley Laboratory and Department of Mathematics. University of California. Berkeley, CA 94720. Vol. 148, pp. 2-22. December, 1997.
3. Calzadilla, F. "Segmentación de Imágenes Médicas Aplicando Texturas". Tesis presentada en opción al título de Licenciado en Ciencias de la Computación. Facultad de Matemática y Computación. Universidad de Oriente. 2009.
4. Carrión, Pilar. "Tecnologías Avanzadas para el Desarrollo de Software Inteligente". Universidad de Vigo. Junio 2006.
5. Gabor, D. (1946). "Theory of Communication". Journal of Institute for Electrical Engineering. 93 (26). 429-457.
6. Laffue, Alfredo. "Anemia Falciforme". Unidad Docente Hospitalaria Instituto de Investigaciones Médicas Alfredo Lanari. Universidad de Buenos Aires, facultad de Medicina. 29 de marzo, 2000.
7. López Álvarez, Xosé L., Pérez Lorenzo, Natalia. "Estudio de una Anemia". <http://www.fisterra.com/guias2/anemia.asp>. Servicio de Atención Primaria Mariñamansa - Ourense- SERGAS- España. 24 de enero, 2005.
8. Pajares, G. y De la Cruz, J. "Visión por Computador: imágenes digitales y aplicaciones". I Edición. RAMA. Madrid 2003.
9. Pons Porras, Aurora. "Desarrollo de Algoritmos para la Estructuración Dinámica de Información y su Aplicación a la Detección de Sucesos". Tesis en opción al título de Doctora en Ciencias. Departamento de Lenguajes y Sistemas Informáticos, Escuela Superior de Tecnología y Ciencias Experimentales. Universidad Jaume I. Castellón, Junio 2004.

10. Riquenes Cutiño, Odalis. "En estudio preclínico medicamento cubano contra la siklema". <http://www.juventudrebelde.cu>. Cuba, 26 de septiembre, 2007.
11. Roca Goderich, Reinaldo. "Temas d Medicina Interna". Tercera Edición. Editorial Pueblo y Educación. Noviembre, 1991. Tomo III, pp. 255-263.
12. Terzopoulos, D. "Matching deformable models to images: Direct and iterative solutions". In Topical Meeting on Machine Vision, Technical Digest Series. Vol. 12. 1987.