



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

47
2EJ

SISTEMA AUTOMATICO PARA EL
CALCULO DEL INDICE MITOTICO
POR EL PROCESAMIENTO
DIGITAL DE IMAGENES:
MIT 2000

FALLA DE ORIGEN

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACION

P R E S E N T A

EDGAR GARDUÑO ANGELES

ASESOR: DR. GABRIEL CORKIDI BLANCO

México, D. F. Mayo 1995





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A mis padres, a mis maestros distinguidos, a la Universidad
Nacional Autónoma de México, al Centro de Instrumentos, a mi asesor Dr. Gabriel Corkidi
y a mis "hijos".*

INDICE

INTRODUCCION	1
1. INDICE MITÓTICO	2
1.1. Metodología de Conteo Manual	6
1.2. Antecedentes	7
2. OBJETIVOS	8
CAPITULO I	9
I.1. PROCESAMIENTO DIGITAL DE IMÁGENES	9
I.1.1. Digitalización de imágenes	9
I.1.2. Mejoramiento de imágenes	9
I.1.3. Restauración	10
I.1.4. Codificación de imágenes	11
I.1.5. Segmentación	12
I.1.6. Representación y Descripción.	12
I.1.6.1. Textura	12
I.2. FUNCIONES DE PDI UTILIZADAS EN EL PROYECTO MIT 2000	13
I.2.1. Digitalización	13
I.2.2. Operaciones espaciales	13
I.2.2.1. LUT's	13
I.2.2.2. Dilatación	15
I.2.2.3. Operaciones de Vecinos (Kernels)	17
I.2.3. Dominio de la frecuencia	19
I.2.4. Segmentación	19
I.2.4.1. Hallazgo de umbral	19
I.2.4.2. Proceso de segmentación	23
I.2.5. Textura	24
CAPITULO II	25
SISTEMA MIT 2000.	25
II.1. Sistema BIOCOM 2000	25
II.1.1. Hardware	26
II.1.2. Software	27
II.1.3. Manejo de memoria	28
II.1.4. Estructura	29

CAPITULO III	32
INTERFAZ CON EL MICROSCOPIO	32
III.1. Conexión física	32
III.2. Controlador del movimiento	33
III.2.1. Hardware	33
III.2.2. Software	33
III.3. Enfoque	35
III.3.1. Contraste	35
III.3.2. Movimiento	36
III.4. BARRIDOS	38
III.4.1. Movimiento	38
III.4.2. Desplazamientos	38
III.4.3. Definición de Areas de Observación	38
III.4.4. Selección del Area de Observación	39
III.4.5. Traslape	40
III.4.6. Limites	40
III.4.7. Zona Valida de Barrido	41
III.4.8. Tipos de Barrido	41
III.4.9. Manejo de Controles y Experimentos.	43

CAPITULO IV	45
CALCULO DEL INDICE MITOTICO	45
IV.1. Categorías	45
IV.1.1. Núcleos	45
IV.1.2. Mitosis	47
IV.2. Representación y Descripción de clases	49
IV.2.1. Segmentación	49
IV.2.2. Obtención de parámetros	50
IV.3. Búsqueda de parámetros que definen a las distintas categorías	51
IV.3.1. Núcleos	51
IV.3.2. Mitosis Compactas	58
IV.3.2.1. Textura	63
IV.3.3. Mitosis Dispersas	67
IV.4. Otras Operaciones	79
IV.4.1. Visualización	79
IV.4.2. Orden de Búsqueda de Parámetros	79
IV.4.3. Orden de Búsqueda de Categorías	80
IV.4.4. Problemas de Segmentación	85
IV.4.5. Entrenamiento	92
IV.4.6. Búsqueda de limites para los parámetros	96
IV.5. Rutina de cálculo del Índice Mitótico	100

IV.6. Rutinas Relacionadas	106
IV.6.1. Rutina leecmd	108
IV.6.1.1. Ejecución del Archivo de Comandos	110
IV.6.1.2. Movimiento de las platinas para leecmd	110
IV.6.2. Rutina mitcmd	110
IV.7. Rutina de verificación	112
IV.7.1. Archivo de Control	112
IV.7.2. Rutina de verificación	114
IV.7.3. Auxilio Extra de Verify	117

CAPITULO V	119
Pruebas y Ajustes del Prototipo	119
V.1. Interfaz con BIOCOM	119
V.2. Interfaz con el MultiControl 2000	119
V.3. Archivos de Control	120
V.4. Rutinas de lectura de Archivos de Comandos	120
V.5. Rutinas de Verificación de Ejecución del Índice Mitótico	120
V.6. Proceso de Reconocimiento	120
V.7. Resultados del Reconocimiento	121
V.8. Errores de Reconocimiento	125
V.8.1. Filtros Discriminantes	132
V.9. Memoria	135

CAPITULO VI	136
Resultados, Perspectivas y Conclusiones	136
Referencias	138
Apéndice A Manual de Operación	140
Apéndice B Archivo de Soporte.	183
Apéndice C Rutinas del sistema.	188
Apéndice D Diccionario de Variables y Constantes	210
Apéndice E Glosario	219
Apéndice F Interfaz C-FORTRAN	221
Apéndice G Listado del programa (disco)	

INTRODUCCIÓN

La ingeniería se refiere a la utilización, invención o perfeccionamiento de herramientas o técnicas que son usadas tanto en la vida diaria como en el área científica. Bajo este principio la ingeniería ha ido coadyuvando al desarrollo de diferentes ramas de la ciencia, este es el caso de la llamada ingeniería biomédica.

La ingeniería biomédica puede definirse como la aplicación de los principios de ingeniería, y sus prácticas, a problemas de biología y medicina para el mejoramiento de la vida humana.

El campo de la ingeniería biomédica es muy amplio, abarca actividades tales como entender y cuantificar funcionamientos normales y anormales, predecir y diagnosticar cambios debidos a alteraciones así como a la proposición y diseño de métodos para enfrentar un fenómeno biomédico.

En las últimas décadas la ingeniería biomédica ha experimentado un acelerado desarrollo con los avances tecnológicos que han ocurrido recientemente en las áreas de la electrónica y computación. La ingeniería biomédica ha crecido a la par de dichos desarrollos. Se observa una interacción y retroalimentación en el desarrollo del conocimiento de estas ciencias, así por ejemplo, los avances en la electrónica y la computación han dado lugar a nuevos desarrollos en campos como la imageonología, sonografía, etcetera, que han impulsado, a su vez, el desarrollo de la ingeniería biomédica y de sus ciencias base, medicina y biología. Es por ello que en la actualidad es necesario ampliar el conocimiento de la ingeniería hacia las ramas biomédicas, aumentando el conocimiento sobre la metodología de investigación, instrumentación y técnicas de medición en dichas áreas del conocimiento.

La Universidad Nacional dedica valiosos esfuerzos para el desarrollo de aplicaciones de ingeniería al área biomédica en varios de sus centros de investigación, tal es el caso del Laboratorio de Procesamiento de Imágenes del Centro de Instrumentos.

En dicho laboratorio se desarrollan algoritmos para el procesamiento digital de imágenes, programas y otras aplicaciones para el área de la biomedicina. Dicho laboratorio colabora con dependencias dentro y fuera de la misma universidad tales como el Instituto de Investigaciones Biomédicas, el Instituto de Fisiología Celular, Instituto de Biotecnología y la Facultad de Ciencias entre otros.

La colaboración del Laboratorio de Procesamiento de Imágenes con dichas dependencias ha arrojado varios proyectos de colaboración interdisciplinaria. Como resultado de este esfuerzo de desarrollo interdisciplinario surge el Proyecto para el Cálculo del Índice Mitótico por medio del Procesamiento Digital de Imágenes (PDI). Dicho

proyecto cuenta con la colaboración del Laboratorio de Citogenética del Instituto de Investigaciones Biomédicas y el Laboratorio de Procesamiento de Imágenes del Centro de Instrumentos.

El Laboratorio de Citogenética del Instituto de Investigaciones Biomédicas realiza, dentro de muchos otros, estudios para el análisis de los efectos de las drogas en la reproducción de las células. Una de las tareas, indispensable para este análisis, es el cálculo del índice mitótico.

El cálculo del índice mitótico es una medida que se utiliza en biología y que se refiere a la relación que existe entre los cuerpos en proceso mitótico y los cuerpos estables en cierta muestra.

En el área de citogenética es necesario realizar este conteo debido a que esto ayuda a ver los efectos de una droga o medicamento sobre la reproducción de las células. En el Laboratorio de Citogenética el cálculo del índice mitótico es aplicado sobre linfocitos, mejor conocidos como glóbulos blancos, en una muestra de sangre. Esta operación es monótona y requiere que el investigador lea las laminillas, lo cual lleva horas de trabajo.

El procesamiento de imágenes ha demostrado ser una poderosa herramienta en el diagnóstico de fenómenos biológicos y en la confirmación de otros fenómenos que antes hubieran requerido de una labor mayúscula por parte de investigadores y técnicos para llegar a las mismas conclusiones.

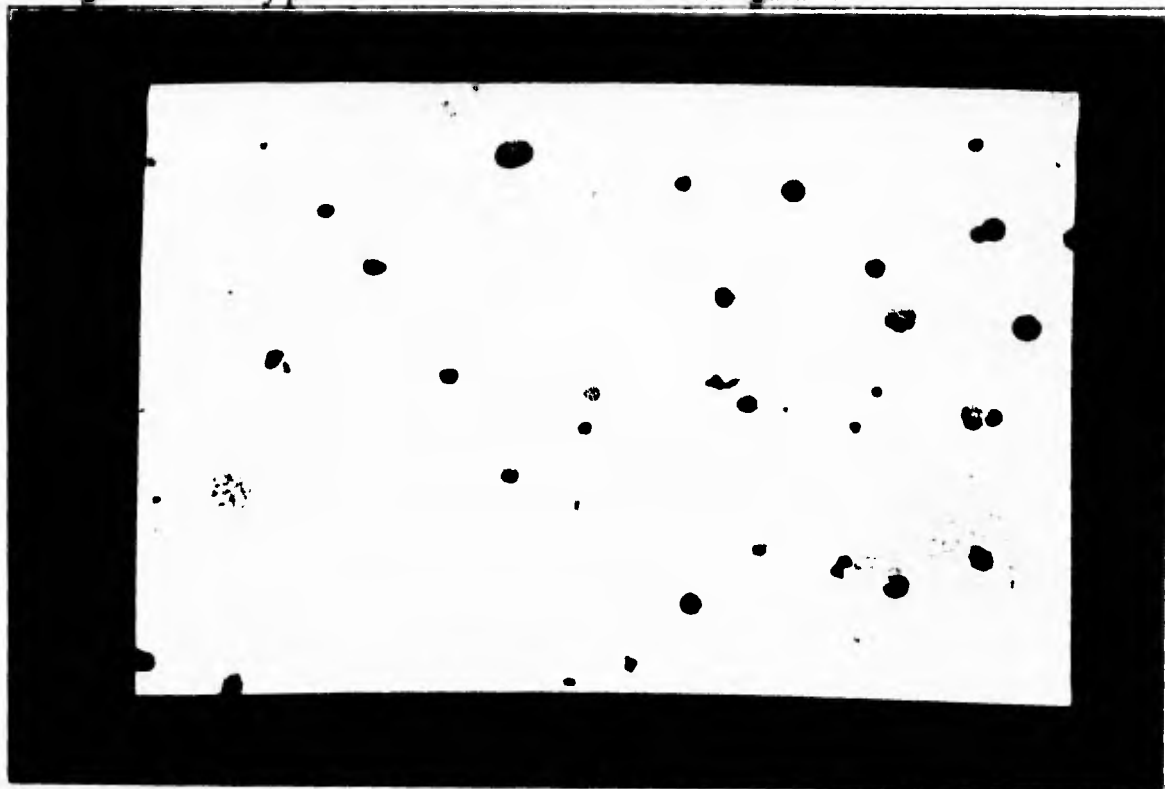
1. Índice Mitótico

Cualquier producto que sea utilizado para mejorar la salud humana, puede tener efectos sobre la proliferación celular. Es por esto que todas las agencias ocupadas de la salud del hombre exigen pruebas de citotoxicidad.

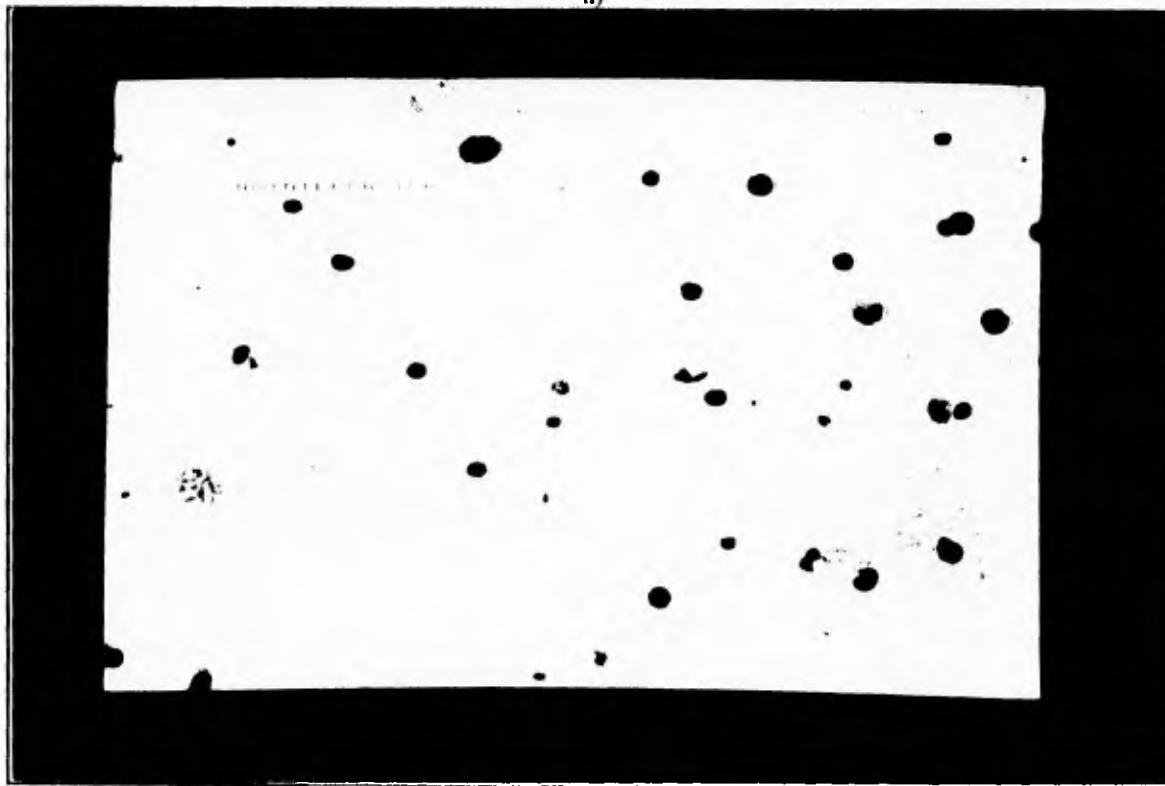
El Índice Mitótico es uno de los parámetros más comunes que se usa para medir la proliferación de las células; indica el porcentaje de células que están en proceso de división. En términos generales, el índice mitótico es un marcador ampliamente utilizado para evaluar y comparar la velocidad de proliferación de las células. Dicho índice puede ser afectado por la exposición a agentes físicos, químicos y biológicos que conllevan a citotoxicidad lo cual resulta en una disminución en la proliferación y consecuentemente en una reducción del índice mitótico.

El analista identifica y cuenta miles de células y reporta el porcentaje de células en mitosis entre las células interfásicas (células que solamente están estimuladas sin llegar a un proceso de división). El reconocimiento de las células interfásicas se realiza de la manera

siguiente: las células interfásicas tienen por lo general una tonalidad típica, entran dentro de un rango de tamaño y poseen una forma casi circular ver figura 1.



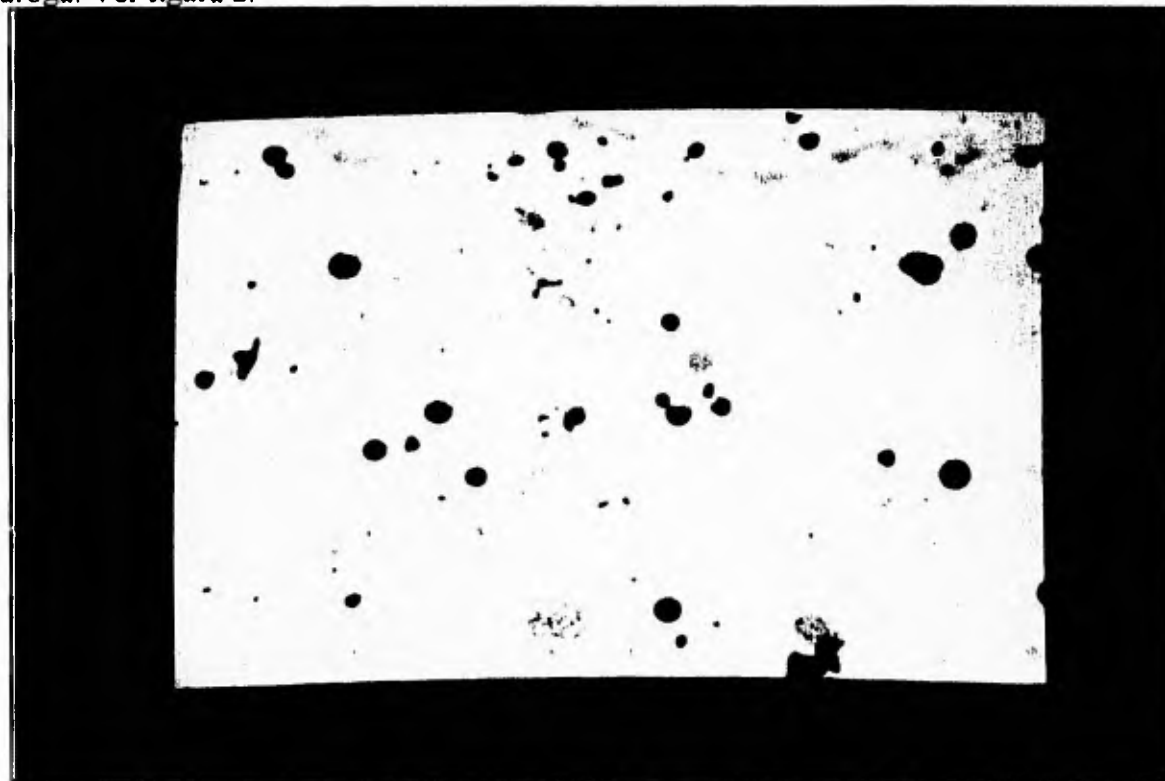
a)



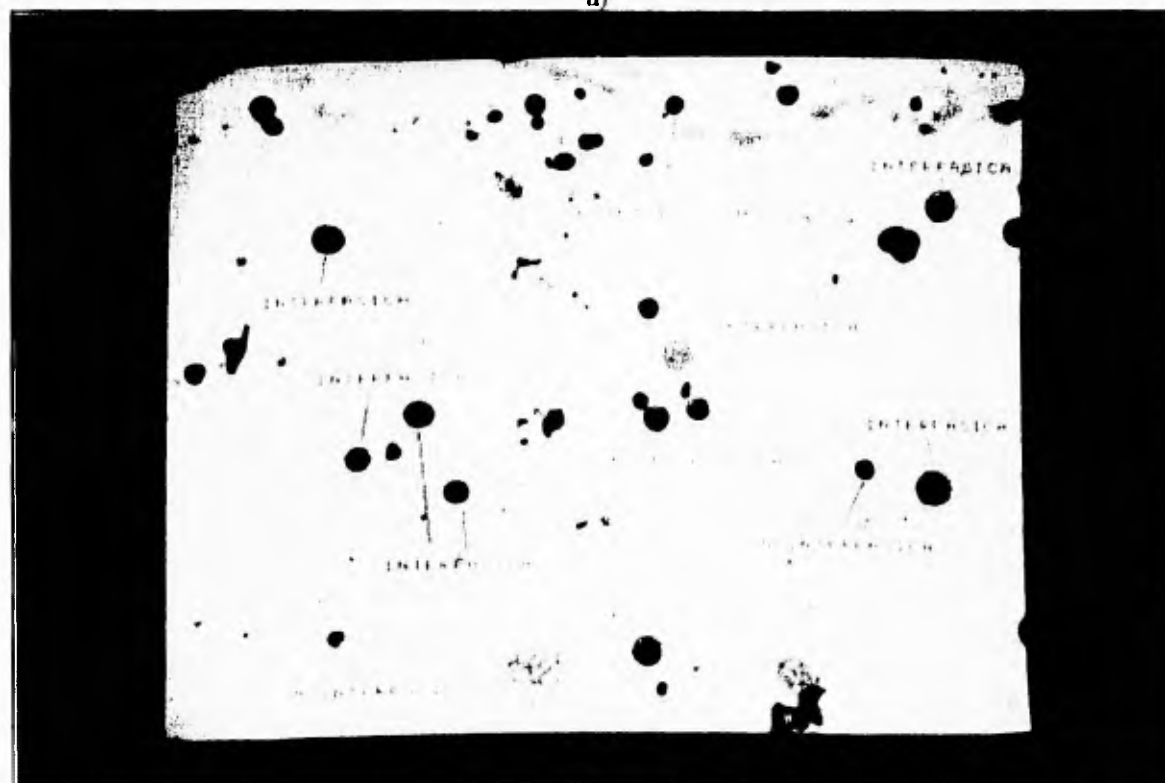
b)

Fig. 1 Células interfásicas a) ejemplo 1. b) ejemplo 2.

En la muestra también se pueden hallar células más pequeñas y de tonalidad más oscura; éstas no entran dentro de la cuenta debido a que no fueron estimuladas por la droga. Ver figura 2.



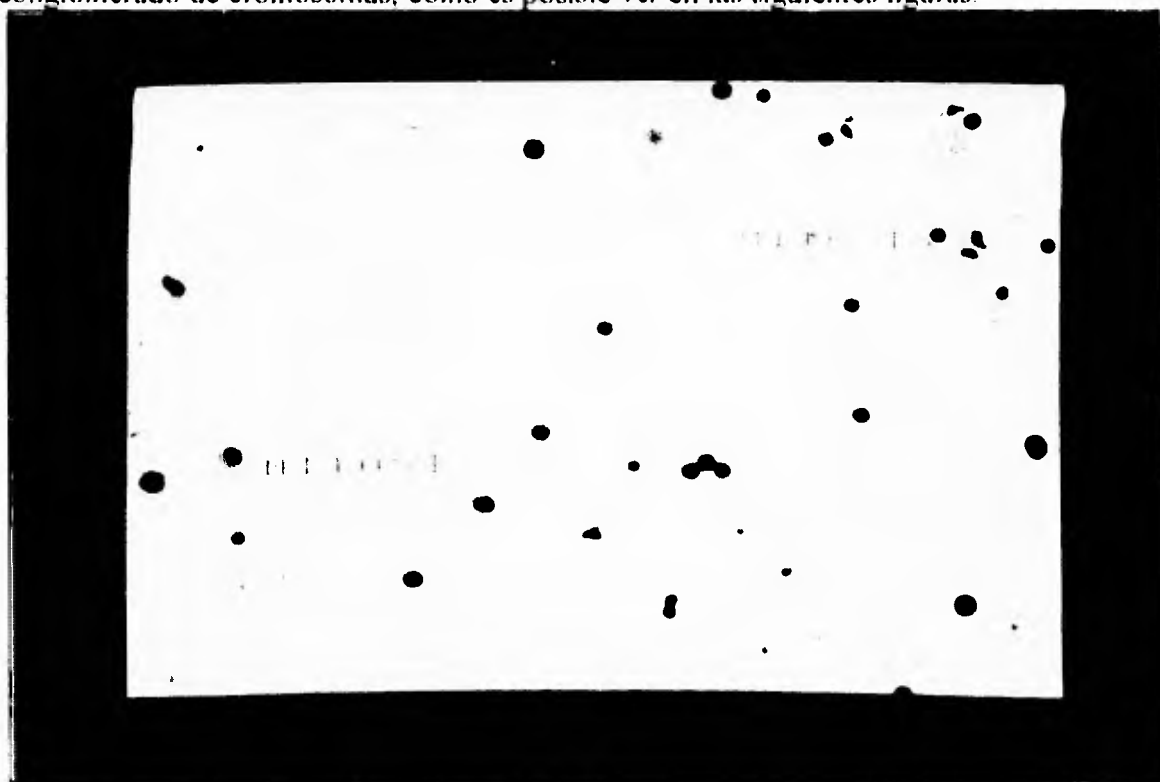
a)



b)

Fig. 2 Células no interfásicas a) ejemplo 1. b) ejemplo 2.

Para las células en mitosis el reconocimiento se basa en la identificación de un conglomerado de cromosomas, como es posible ver en las siguientes figuras.



a)



b)

Fig. 3 mitosis a) ejemplo 1 b) ejemplo 2 c) ejemplo 3



c)

Fig. 3 mitosis. a) ejemplo 1. b) ejemplo 2. c) ejemplo 3.

Para este conteo los analistas se basan en su experiencia adquirida después de contar varios portaobjetos, para clasificar los diferentes cuerpos antes mencionados y, sobre todo, poder discernir entre las células que están en el límite de dos diferentes categorías.

Tradicionalmente se analizan las preparaciones al microscopio de luz. Para tener un resultado estadísticamente significativo del índice mitótico en una laminilla, comúnmente se hace un conteo de 2000 elementos que incluyen células en mitosis y células interfásicas. A partir de esos elementos se calcula la relación entre células en mitosis e interfásicas. La evaluación de la citotoxicidad de una sustancia requiere un mínimo de 40 horas promedio de trabajo de observación al microscopio por personal especializado. Es un proceso largo y tedioso. Indudablemente, la automatización de dicho proceso podrá traer beneficios enormes en el área de citogenética.

1.1. Metodología de Conteo Manual.

Una vez hecha la preparación del cultivo que se va a analizar, se prosigue a verterla a las laminillas (las laminillas que son utilizadas son del tipo esmerilado). Cabe mencionar que un experimento puede componerse de más de un portaobjetos, sin embargo no siempre es necesario revisar todas las laminillas. Antes de revisar todo el experimento se realiza una

revisión de controles, es decir una o dos laminillas que servirán como muestra de la preparación. Si las laminillas-contrales poseen un Índice Mitótico mayor a un cierto porcentaje (típicamente 5%) entonces se supone que la preparación es apta para revisarse y se prosigue a revisar todas las laminillas del experimento, esto se debe a que este porcentaje es un indicador de que la preparación está bien hecha.

La revisión manual se lleva a cabo comenzando por el lado esmerilado. Moviéndose en forma paralela al extremo más corto, se va realizando la cuenta de los cuerpos. Al terminar con dicha línea se avanza un campo por el lado más largo y se vuelve a barrer una línea paralela al extremo más corto solo que en sentido contrario. Este movimiento de zigzag se continua hasta acabar con la laminilla o hasta alcanzar el número de cuerpos que se desean contar:

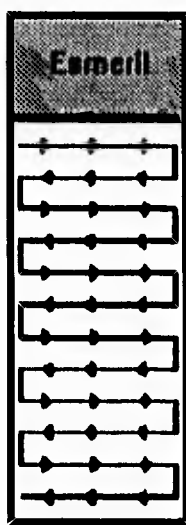


Fig. 1.4 Barrido de las laminillas.

1.2. Antecedentes

Se han desarrollado algunos sistemas para automatizar este proceso, pero hasta el momento todos estos son únicamente una ayuda para un conteo interactivo y semiautomático, o no dan el resultado específico del índice mitótico (16, 17, 18, 19, 20, 3). La mayoría de los sistemas, como el llamado buscador de metafases, han sido implementados para la detección rápida de buenas metafases en la que los cromosomas estén bien distribuidos y puedan ser bien cariotipados. También se ha desarrollado equipo sofisticado que basados en la morfología, fotometría y parámetros de textura permiten reconocer células en diferentes estados del ciclo celular, pero requieren fotomultiplicadores y microfotómetros, además de la computadora y el microscopio.

En años recientes se ha propuesto el uso de fluorocitómetros que distinguen las diferentes fases del ciclo celular con base al contenido de DNA; ellos son ayudados con la

información que se obtiene del histograma celular pero no reconoce las figuras mitóticas. Existe cierto trabajo orientado para los patólogos, pero las cuentas mitóticas se realizan manualmente.

Ninguno de los sistemas arriba mencionados reconoce simultáneamente células y metafases, lo cual es necesario para estimar el índice mitótico en los cultivos celulares. Se puede afirmar que no existe en la actualidad ningún equipo que permita el análisis del índice mitótico de forma automática, debido tanto a la complejidad y variabilidad de los objetos a cuantificar; como ejemplo basta mencionar que las células en división pueden cambiar de tamaño y forma, lo que las puede confundir con otros artefactos.

2. Objetivos.

El objetivo de este trabajo es lograr la creación de un sistema automatizado para la evaluación cuantitativa de la proliferación linfocitaria, que llamaremos MIT 2000. Este sistema tiene por objeto reducir el tiempo que gastan los investigadores en el cálculo de la proliferación linfocitaria, tiempo que podría ser usado en otras tareas de investigación en el área de investigación biomédica básica y farmacológica.

CAPITULO I

1.1. PROCESAMIENTO DIGITAL DE IMÁGENES

El interés en el procesamiento de imágenes nació debido a dos posibles aplicaciones: mejoramiento de la información pictórica para la interpretación humana y el procesamiento de datos escénicos para la percepción autónoma de máquinas. Las técnicas para el procesamiento de imágenes pueden ser divididos en cuatro categorías principales:

1. Digitalización de imágenes.
2. Mejoramiento y restauración de imágenes.
3. Codificación de imágenes.
4. Segmentación de imágenes.
5. Representación y descripción de imágenes.

El termino imagen se refiere a una función bidimensional de intensidad de luz, denotada por $f(x,y)$, donde el valor o amplitud de f en las coordenadas espaciales (x,y) da la brillantez de la imagen en dicho punto.

La intensidad de una imagen monocromatica f en las coordenadas (x,y) es conocida como nivel de gris de la imagen en dicho punto.

El presente trabajo hace uso de las técnicas de digitalización, mejoramiento de imágenes, segmentación de imágenes y representación y descripción por lo que serán explicadas brevemente. Las técnicas de restauración y codificación se explican por completez.

1.1.1. Digitalización de imágenes

Para que sea posible que las imágenes sean procesadas por una computadora, la función $f(x,y)$ debe ser digitalizada espacialmente y en su amplitud. La digitalización en las coordenadas espaciales (x,y) se refiere al muestreo de la imagen, mientras que la digitalización de la amplitud se conoce como cuantización del nivel de gris.

La digitalización de una imagen requiere que se haga una decisión sobre el número de componentes que tendrá la imagen, el número de pixeles, y del valor en gris que tendrá cada pixel. Como puede verse esto solo es una aproximación de la imagen contigua(6).

1.1.2. Mejoramiento de imágenes

El principal objetivo de las técnicas de mejoramiento es procesar una imagen dada de forma tal que el resultado sea mejor para una aplicación en particular. Normalmente los métodos para el mejoramiento de una imagen se pueden dividir en dos: métodos sobre el dominio de la frecuencia y métodos en el dominio espacial.

Los métodos espaciales se refieren a aquellos procedimientos que operan directamente sobre los píxeles (6,7). Esto se puede expresar de la manera siguiente:

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

donde $f(x,y)$ es la imagen de entrada, $g(x,y)$ es la imagen resultante y T es un operador sobre f definido sobre los vecinos de (x,y) .

El principal acercamiento usado para definir un vecino sobre (x,y) es usar una imagen cuadrada o rectangular centrada en (x,y) . La forma más simple de T es de un solo casillero pues solo se hace un mapeo de f , resultando $g=f$.

En el caso de las técnicas en el dominio de la frecuencia estas se basan en el teorema de convolución:

$$g(x,y) = h(x,y) * f(x,y)$$

por el teorema de convolución se tiene:

$$G(u,v) = H(u,v)F(u,v)$$

donde G , H y F son las transformadas de Fourier de g , h y f , respectivamente. En una aplicación típica de mejoramiento se tiene $f(x,y)$ y se necesita seleccionar $H(u,v)$ de tal forma que la imagen deseada, dada por:

$$g(x,y) = F^{-1}[H(u,v)F(u,v)]$$

exhiba alguna característica de la función $f(x,y)$ en $g(x,y)$ por medio de hallar la función $h(x,y)$ o $H(u,v)$.

Algunos ejemplos de las técnicas de mejoramiento son:

- Ecuilización por medio del histograma
- Filtrado usando un filtro pasa-bajas
- Filtrado usando un filtro pasa-altas
- Filtrado usando un filtro pasa-bandas
- Afinamiento de imágenes.

1.1.3. Restauración

El propósito de la restauración de una imagen es el de mejorar la imagen en algún sentido. Para hacer la diferencia entre mejoramiento y restauración se considera a esta última como el proceso que intenta reconstruir o recobrar una imagen que ha sido degradada, usando un conocimiento anterior sobre el fenómeno que la degrado. Estas técnicas están orientadas hacia el modelado de la degradación y a aplicar el proceso inverse que provocó la degradación para recuperar la imagen. Esto requiere de un criterio satisfactorio para decir que una imagen está bien reconstruida.

Por otra parte, las técnicas de reconstrucción, normalmente, son procedimientos heurísticos que fueron diseñados para manipular una imagen tomando ventaja de los aspectos psicofísicos del sistema visual humano(6).

De esta forma el proceso se puede formular como una imagen original f que fue sometida a un proceso H dando como resultado la imagen g .

$$g=Hf$$

Se debe tomar en cuenta el efecto del ruido n en este proceso, quedando como:

$$g=n+Hf$$

entonces la reconstrucción quedaria como:

$$f=H^{-1}g \quad \text{o} \quad f=H^{-1}(g-n)$$

lo cual puede expresarse de la siguiente forma:

$$f(x, y) = F^{-1} \left[\frac{G(u, v) - N(u, v)}{H(u, v)} \right]$$

Algunos ejemplos de la restauración son:

- Filtro Wiener
- Restauración restringida a mínimos cuadrados
- Restauración interactiva
- Interpolación de niveles de grises

1.1.4. Codificación de imágenes

Las representaciones digitales de las imágenes normalmente requieren de un gran número de bits. Para algunas aplicaciones es necesario representar las imágenes con el menor número de bits posibles.

La codificación de imágenes normalmente se divide en tres categorías: compresión de datos, transmisión de imágenes y extracción de características. Estas técnicas están orientadas a la resolución de problemas específicos, es decir la técnica usada depende del problema que se tenga (6).

Para este tipo de técnicas es necesario tener en cuenta criterios de fidelidad y de cuantificación de los datos.

El proceso de codificación queda dividido en las siguientes etapas:

1. Mapeo de la imagen.
2. Cuantificación de la imagen
3. Codificación por medio de métodos como:
 - 3.1. Entropía
 - 3.2. Huffman
 - 3.3. Códigos B
 - 3.4. Códigos de corrimiento

1.1.5. Segmentación

La segmentación es el proceso que subdivide una imagen en sus partes constituyentes u objetos. La segmentación es uno de los elementos más importantes en el análisis automatizado de imágenes debido a que al realizar este paso se tienen objetos o entidades de interés que pueden ser descritos o reconocidos posteriormente. Es pertinente decir que muchos errores de análisis de los objetos parten de una mala segmentación de la imagen.

Los algoritmos de segmentación generalmente están basados en una o dos propiedades de los valores de grises: discontinuidad y similitud. Para la primera categoría se parte la imagen basándose en cambios abruptos en los niveles de grises. Las principales áreas de interés dentro esta categoría son la detección de puntos aislados y la detección de líneas y bordes en una imagen. La segunda categoría está basada en encontrar un umbral, el crecimiento de una región y en la división y unión. El concepto de segmentación de una imagen basándose en la discontinuidad o similitud de los valores de grises de sus píxeles es aplicable a imágenes estáticas o dinámicas(6).

1.1.6. Representación y Descripción.

Después de que una imagen ha sido segmentada en regiones por los métodos deseados normalmente se desea representar y describir el resultado de los píxeles agrupados en una forma que se ajuste a un procesamiento de computadora posterior. Existen dos formas básicas de representar una región: a) basándose en sus características externas y b) basándose en sus características internas. La representación de un grupo de píxeles es la parte inicial para el procesamiento de dicho grupo, como segunda parte se debe describir la región basándose en el esquema seleccionado para su representación.

Generalmente la representación externa se elige cuando se desea hablar de las características de forma de los objetos (morfología) mientras que la representación interna es elegida cuando uno se interesa en las propiedades reflectivas (color, textura, etcetera). En cualquiera de los dos casos es importante que los descriptores sean insensibles a los cambios en tamaño, traslación y rotación(6).

1.1.6.1. Textura

Un acercamiento importante a la descripción de la región es cuantificar el contenido de textura. Sin embargo no existe una definición precisa de lo que es textura sin embargo puede considerarse como una medida de lo que es la suavidad, la regularidad y la rugosidad de un objeto.

Hay tres acercamientos a la textura en procesamiento de imágenes: estadístico, estructural y espectral. El estadístico produce caracterizaciones de textura tales como

suavidad, granularidad, rugosidad, etcetera. Las técnicas estructurales trabajan con líneas paralelas separadas espacialmente. Las técnicas espectrales están basadas en las propiedades del espectro de Fourier y son usadas principalmente para detectar la periodicidad global en una imagen por medio de la identificación de los picos agudos de alta energía en el espectro(6).

I.2. FUNCIONES DE PDI UTILIZADAS EN EL PROYECTO MIT 2000

En el proyecto MIT 2000 se usan algunas operaciones de procesamiento de imágenes que se mencionan a continuación:

I.2.1. Digitalización

Para el proyecto es necesario realizar la digitalización de imágenes pero esto es dejado al hardware del sistema (Ver en el capítulo dos la explicación de la tarjeta MATROX) que maneja imágenes digitales de 512x480 píxeles en 256 niveles de gris.

La forma en que el hardware obtiene estas imágenes es haciendo pasar la señal de video por un convertidor A/D de tiempo real que produce píxeles de ocho bits, posteriormente por las tablas de asignación de entrada para luego continuar a un *buffer* de cuadro. Hasta aquí la imagen ha sido digitalizada para su manipulación pero la imagen debe ser desplegada en el monitor "imagen" para que el usuario pueda visualizar los tratamientos sobre la imagen. Para esto la señal proveniente del *buffer* de cuadro pasa por un video codificador, después pasa por las tablas de asignación de salida para seguir por el convertidor D/A de tiempo real.

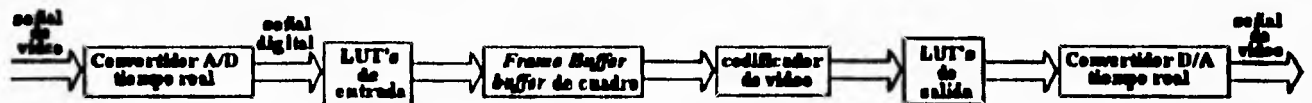


Fig 1.1 Esquema de la digitalización por la tarjeta MATROX.

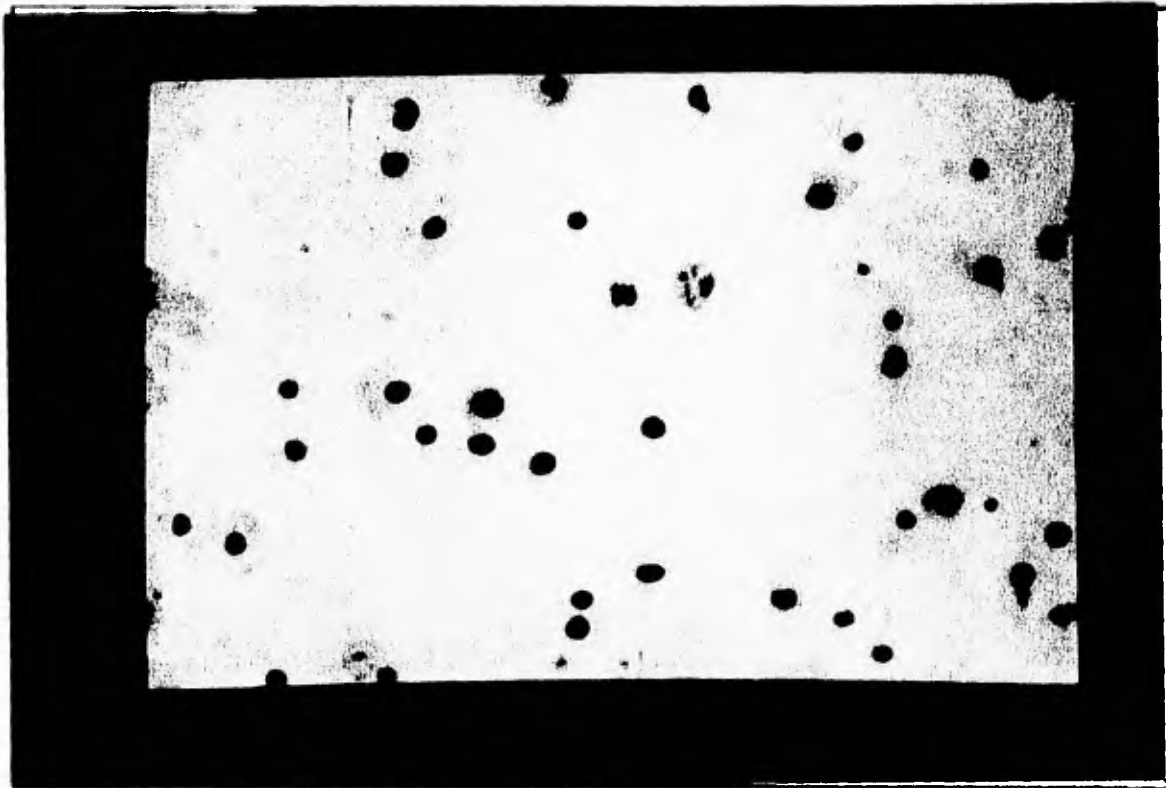
I.2.2. Operaciones espaciales

I.2.2.1. LUT's

La relación entre los valores de brillantez original y de reemplazo se le conoce como función de transferencia, y puede alterar la apariencia de una imagen. Esta función no es necesariamente lineal; de hecho las funciones de transferencia no lineales son bastante usadas. Una función de transferencia logarítmica mimetiza la forma en que la mayoría de las

películas fotográficas trabajan y puede ser descrita por un valor γ que da la pendiente de una gráfica semi-logarítmica.

El uso de tablas de búsqueda dentro de la imagen LUT's (tablas de asignación o *Look Up Tables* en inglés) para implementar una función de transferencia es mucho más rápido y completamente no-destructivo. Usando LUT's, la función de transferencia por si misma es escrita sobre el *hardware*. Esto hace muy fácil el cambiar la apariencia del despliegue. El ojo humano es, en general, capaz de distinguir tan solo un reducido número de niveles de gris, quizás de veinte a treinta bajo buenas condiciones. Modificando las LUT's, es posible modificar el contraste y/o la brillantez y lograr, así, mejorar la apreciación de la imagen (fig 1.2).



a)

Fig. 1.2 Aplicación de LUT's. a) Imagen original. b) Aplicación de LUT's (resultado).



b)

Fig. 1.2 Aplicación de LUT's. a) Imagen original. b) Aplicación de LUT's (resultado).

En la mayoría de los sistemas de procesamiento de imágenes las LUT's están implementadas en *hardware*. Generalmente, una serie de LUT's están localizadas entre el convertidor analógico-digital (A/D) de entrada y el *buffer* de trabajo y otras entre el *buffer* de trabajo y convertidor digital-analógico (D/A) de salida de la imagen. Normalmente se usan varias LUT's, una para cada color primario, azul, rojo y verde.

El uso de LUT's para el despliegue es distinto al uso de LUT's para la entrada de imágenes. Algunas tarjetas para la adquisición de video permiten el uso de LUT's para la entrada lo que permite realizar operaciones puntuales antes de que la imagen sea almacenada en el *buffer* de trabajo. El uso de LUT's de entrada puede ser de valor cuando la precisión de la digitalización es mayor que la precisión del almacenamiento.

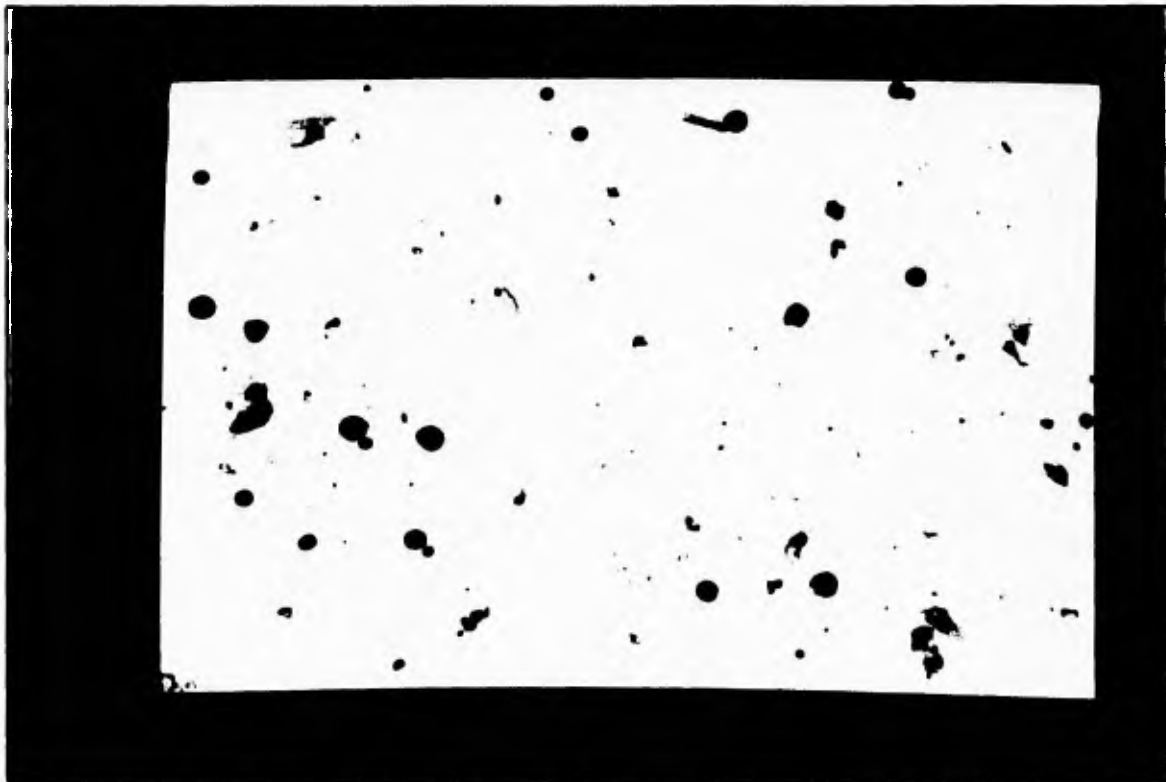
1.2.2.2. Dilatación

La dilatación también se considera un operador morfológico, debido a que es capaz de modificar la forma de un objeto.

Si se aplica una máscara a cada pixel de la imagen siguiendo el principio:

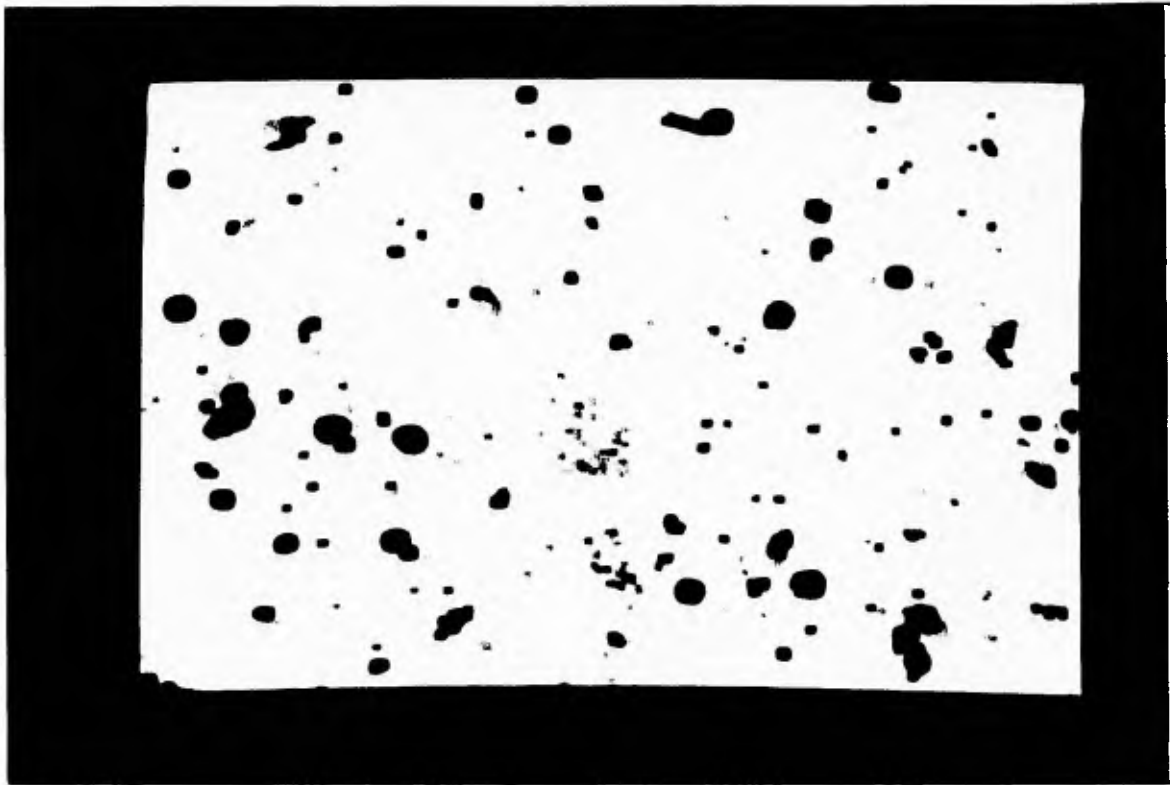
$$G = \bigvee_{k=-R}^R \bigvee_{l=-R}^R M_{k,l} \wedge G_{m-k, n-l}$$

donde \vee es la operación lógica OR y \wedge la operación lógica AND, la imagen G es convolucionada con una máscara M simétrica $2R+1, 2R+1(7,8)$. Ver figura 1.3.



a).

Fig. 1.3 Proceso de dilatación. a)Imagen original. b) Imagen dilatada.



b)
Fig. 1.3 Proceso de dilatación. a) Imagen original. b) Imagen dilatada.

1.2.2.3. Operaciones de Vecinos (Kernels)

En ciertos métodos es necesario hacer uso del valor del pixel a procesar y de sus vecinos(6,7). Por el costo en velocidad y tiempo, normalmente se usan kernels (o máscaras) de 3x3 pixeles aplicando la formula:

$$g(x,y) = \sum_i^m \sum_j^n h(i,j) \cdot f(x-i,y-j)$$

donde $f(x,y)$ es la imagen original, $h(m,n)$ es la máscara y $g(x,y)$ es la imagen resultante. Que resulta de la ecuación en frecuencia:

$$G(u,v) = H(u,v) \cdot F(u,v)$$

La afinación (sharpening en inglés) de la imagen es un método para resaltar los bordes de los objetos y aumentar el contraste de una imagen. En el sistema MIT 2000 se usan máscaras para la función de afinación. Si se considera un kernel de la forma:

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{vmatrix}$$

al aplicarlo a una imagen original se obtiene una imagen de salida cuyos bordes, líneas y puntos discontinuos aparecen, en cambio las áreas uniformes están suprimidas (fig. 1.4). Este proceso mejora la imagen para el ojo humano debido a que el sistema visual humano se fija mejor en los bordes y los operadores de visión temprana dentro de la retina y selecciona dichas discontinuidades para transmitir las al cerebro. Este tipo de operadores se usan para controlar el contraste.

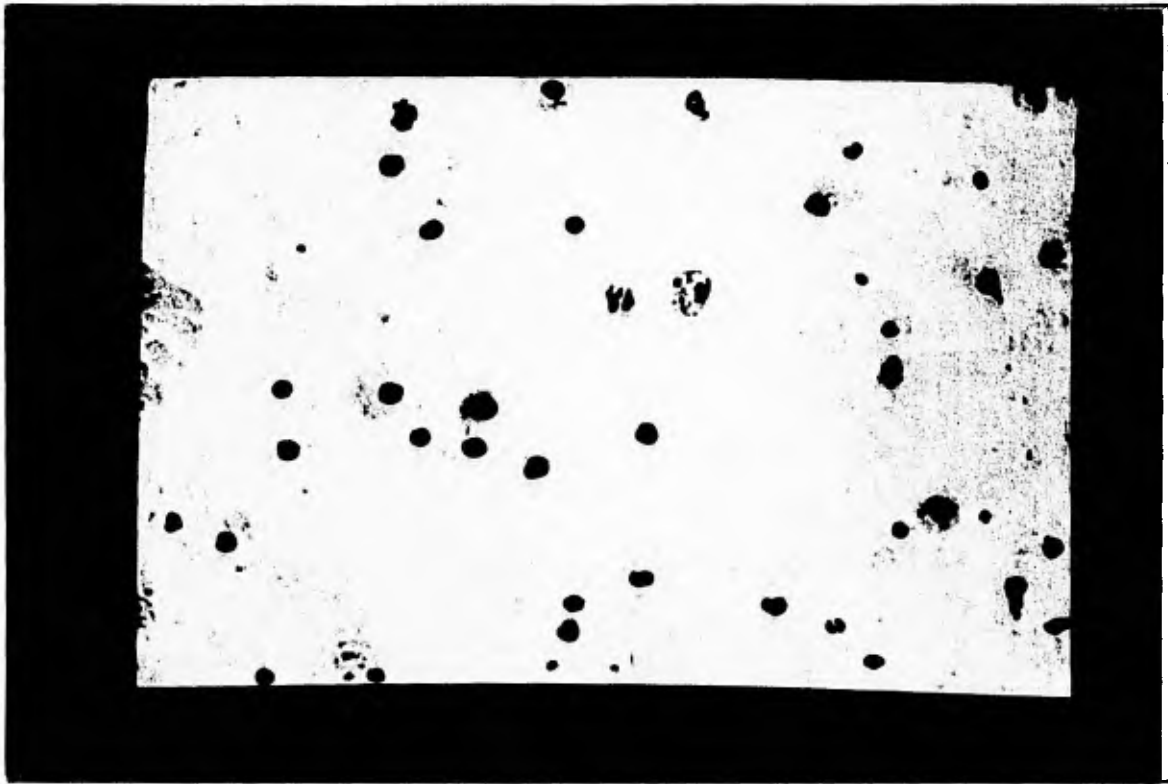


Fig. 1.4 Imagen afinada

Otro tipo de operadores para la afinación de imágenes son:

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix} \qquad \begin{vmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{vmatrix}$$

Además de los operadores anteriores, existen los operadores morfológicos que básicamente trabajan sobre objetos en la pantalla.

1.2.3. Dominio de la frecuencia

El uso de transformadas de frecuencia son costosas en tiempo y memoria. Para una imagen típica de 8 bits de profundidad, una transformada de frecuencia puede requerir al menos 32 bits. Además de requerir una cantidad considerable de memoria necesitan de una mayor cantidad de tiempo para llevarse a cabo, es por ello que en el sistema MIT 2000 no se usarán.

1.2.4. Segmentación

1.2.4.1. Hallazgo de umbral

En la pantalla se pueden identificar varias regiones de interés pero básicamente se puede hablar del fondo de la imagen y objetos de interés, las cuales definen dos poblaciones representadas en la pantalla.

La discriminación por brillantez es actualmente una forma muy usada de seleccionar píxeles que pertenecen a regiones de interés. Este método es muy eficiente comparado con los métodos que usan la detección de límites o el crecimiento de regiones. La discriminación mediante valores de umbral (binarización o *thresholding* en inglés) es una área de gran estudio, pues no siempre se tienen dos regiones de interés bien definidas dentro de la pantalla.

Un método de discriminación por brillantez fue desarrollado por Nobuyuki Otsu(2) basándose en el uso del histograma de la región de interés, normalmente la imagen completa. Este método nos permite evadir la presencia de un operador que realice la operación de hallar el umbral óptimo para dividir las regiones de interés. Esta característica es de gran importancia en sistemas automatizados.

En una situación ideal, el histograma de niveles de gris de una imagen tiene un valle profundo entre dos picos que representan a los objetos y el fondo; así el umbral puede ser elegido como el fondo del valle. Sin embargo, para la mayoría de las imágenes es difícil detectar el fondo del valle, especialmente en casos cuando el valle es ancho y plano o bien cuando hay ruido o cuando los picos difieren mucho en altura, impidiendo que se produzca un valle definido.

El método de Otsu está basado en el análisis discriminante(1,2). En este método, la operación de *thresholding* en una imagen con L niveles de grises, está considerada como una partición de los píxeles en dos clases C_0, C_1 a un nivel de gris k . Esto es: $C_0 = \{0, 1, \dots, k\}$ y $C_1 = \{k+1, k+2, \dots, L\}$. El número de píxeles en el nivel i se denota por n_i y el número total de píxeles expresado por $N = n_1 + n_2 + \dots + n_L$. Se normaliza el histograma de niveles de gris y se considera como una distribución de probabilidad:

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0 \quad \sum_{i=1}^L p_i = 1$$

Por lo tanto, las probabilidades de ocurrencia de clase y los niveles medios de clase, respectivamente, están dados por:

$$\omega_0 = \Pr(C_0) = \sum_{i=1}^k p_i = \omega(k)$$

$$\omega_1 = \Pr(C_1) = \sum_{i=k+1}^L p_i = 1 - \omega(k)$$

y

$$\mu_0 = \sum_{i=1}^k i \Pr(i/C_0) = \sum_{i=1}^k \frac{i p_i}{\omega_0} = \frac{\mu(k)}{\omega(k)}$$

$$\mu_1 = \sum_{i=k+1}^L i \Pr(i/C_1) = \sum_{i=k+1}^L \frac{i p_i}{\omega_1} = \frac{\mu_T - \mu(k)}{1 - \omega(k)}$$

donde

$$\omega(k) = \sum_{i=1}^k p_i, \quad \mu(k) = \sum_{i=1}^k i p_i$$

son los momentos acumulativos de orden cero y primer orden del histograma hasta el nivel k , y

$$\mu_T = \mu(L) = \sum_{i=1}^L i p_i$$

es la media total de la imagen original.

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T, \quad \omega_0 + \omega_1 = 1$$

Las varianzas están dadas por

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 p_i / \omega_0$$

$$\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 p_i / \omega_1$$

Para evaluar que tan bueno es el umbral (en el nivel k) existen los criterios de medición discriminantes:

$$\lambda = \sigma_B^2 / \sigma_W^2, \quad \kappa = \sigma_T^2 / \sigma_W^2, \quad \eta = \sigma_B^2 / \sigma_T^2$$

donde

$$\sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 \text{ varianza dentro de la clase}$$

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \text{ varianza entre-clase}$$

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i \text{ varianza total de niveles}$$

tomando en cuenta las siguientes identidades:

$$\kappa = \lambda + 1, \eta = \lambda(\lambda + 1) \text{ y } \sigma_W^2 + \sigma_B^2 = \sigma_T^2$$

se obtiene el umbral óptimo al hacer máximo η , o en equivalencia, hacer máximo σ_B^2 . El máximo σ_B^2 se selecciona en una búsqueda secuencial usando cantidades acumulativas $\omega(k)$ y $\mu(k)$:

$$\eta(k) = \sigma_B^2(k) / \sigma_T^2$$

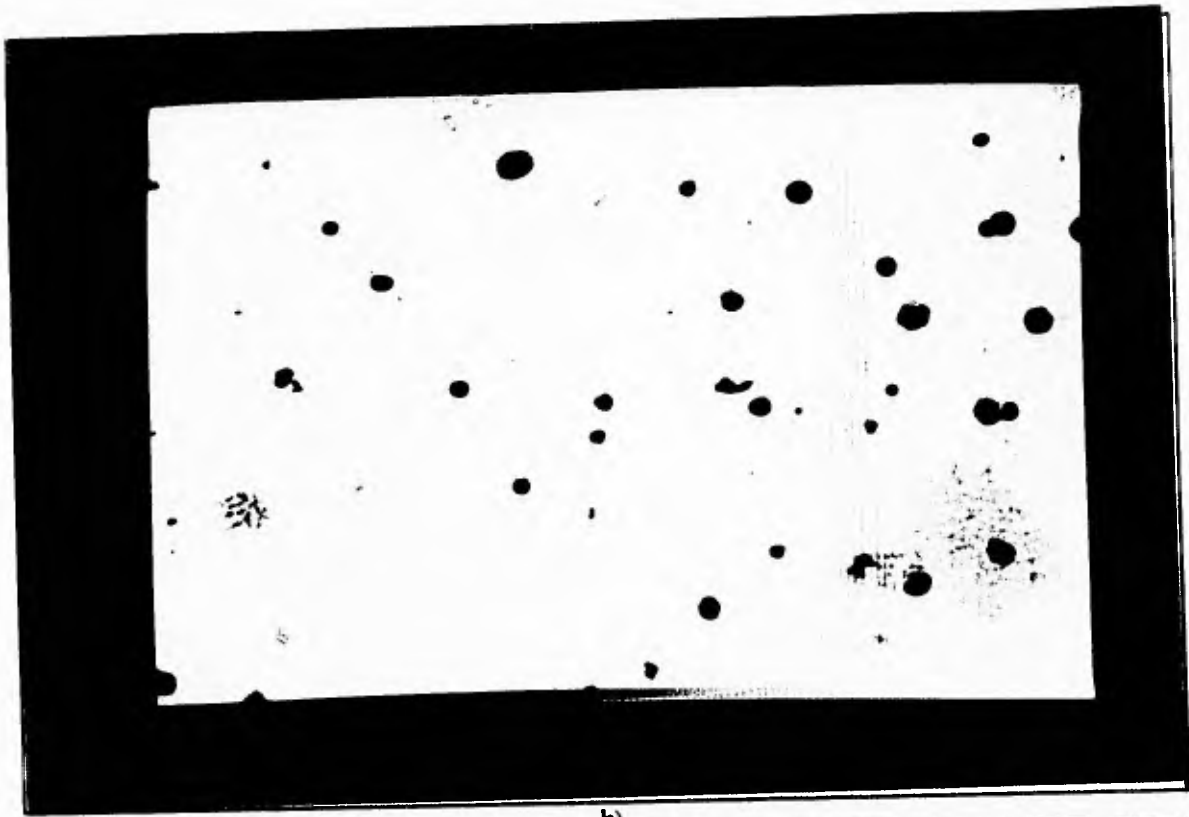
$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}$$

y el valor de umbral óptimo será: $\sigma_B^2(k^*) = \max \sigma_B^2(k)$



a)

Fig. 1.4. Proceso de Segmentación. a) imagen original. b) histograma.
c) histograma. d) resultado del método Otsu.

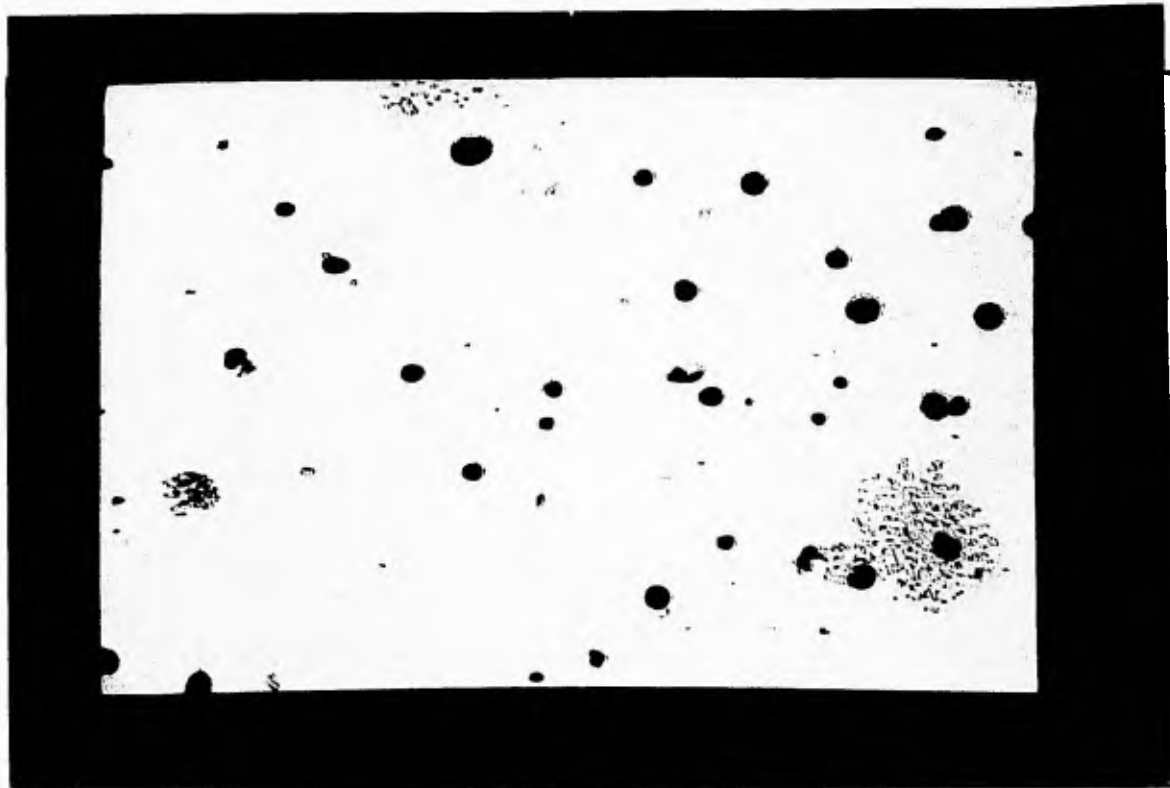


b)



c)

Fig. 1.4. Proceso de Segmentación. a) imagen original. b) histograma.
c) histograma. d) resultado del método Otsu.



d)

Fig. 1.4. Proceso de Segmentación. a) imagen original. b) histograma.
c) histograma. d) resultado del método Otsu.

1.2.4.2. Proceso de segmentación

Una vez que se tiene el nivel de gris para la división de clases es cuestión de implementar otro método para encontrar los límites de cada región. Esto ha sido dejado al sistema BIOCUM(13), ver figura 1.4. Las rutinas del programa realizan la segmentación usando el método de detección de borde (*edge detection*).

El método define a un borde como los límites entre dos regiones con propiedades de niveles de grises relativamente distintas. Se asume que las regiones son lo suficientemente homogéneas de forma que la transición entre dos regiones puede ser determinada sobre la base de las discontinuidades de los niveles de grises.

La forma en que opera el método es haciendo el cómputo de un derivador local, como se puede ver en la siguiente figura:

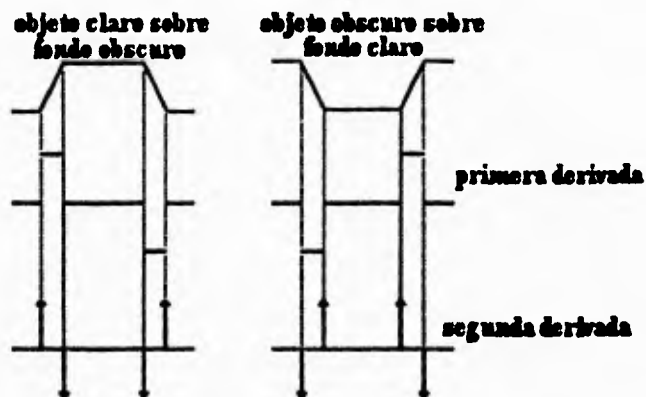


Fig 1.3 Operador local para encontrar los bordes de un objeto claro y uno oscuro.

En el caso de BIOCOTM la imagen es barrida de manera horizontal para encontrar los bordes de los objetos usando el nivel de Otsu como objeto oscuro sobre fondo claro y a veces como objeto claro sobre fondo oscuro.

Con la operación anterior se tiene una imagen que tiene resaltados los bordes de cada objeto ahora hay que saber que borde forma un objeto y cuales no. Para hacer esto BIOCOTM hace una unión de los bordes tomando un punto de la pantalla que representa un borde y va siguiendo a sus vecinos hasta cerrar el perímetro de la superficie, dando como resultado un objeto.

Una vez que se tienen objetos identificados en la imagen, entonces podemos realizar medidas morfológicas sobre dichos objetos, como:

- El área en píxeles de los objetos.
- El nivel de gris medio de los objetos.
- La desviación del nivel de gris de los objetos.
- El centro de gravedad de los objetos.
- La orientación de los objetos con respecto a su posición en la imagen.
- etcetera.

1.2.5. Textura

En el proyecto MIT 2000 se usan identificadores de textura estadísticos. Algunas de las técnicas son los momentos del histograma de niveles de grises de cada objeto como el valor medio de niveles de grises y la varianza o desviación estándar de los niveles de grises. Otros de los métodos empleados serán explicados en el capítulo siguiente cuando se hable de la búsqueda de los parámetros que caracterizan a los objetos de interés.

Una ventaja más que ofrece el sistema BIOCUM para el usuario es que puede operar bajo una interfaz gráfica o por línea de comandos, permitiendo al usuario elegir las rutinas que desea. Como un ejemplo puede observarse la siguiente imagen.

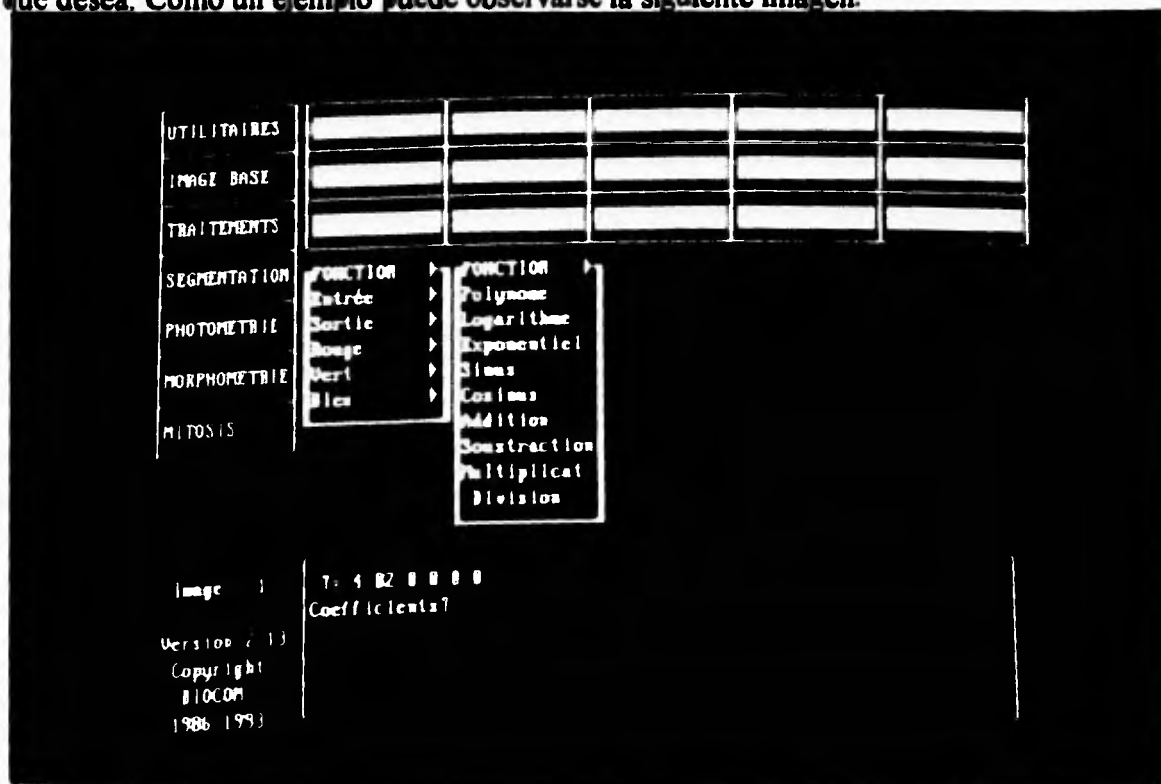


Fig. 2.1 Pantalla de BIOCUM

II.1.1. Hardware

Muchas de las rutinas que ofrece el sistema BIOCUM son posibles debido a que se cuenta con *hardware* especializado capaz de realizar dichos procesos en un tiempo considerablemente corto. Dicho *hardware* es una tarjeta diseñada para el procesamiento de imágenes por la compañía canadiense MATROX. En los diferentes modelos que produce la compañía canadiense siempre se cuenta con una biblioteca de rutinas que puede ser llamada desde C y/o FORTRAN(12).

MATROX fabrica tarjetas digitalizadoras de video que permiten que las computadoras PC puedan realizar operaciones "frame grabbing" (cuadro fijo) sobre una señal de video proveniente de una fuente externa. Las especificaciones de este tipo de tarjetas para PC son:

1. Conexión directa a una bahía de la tarjeta principal de la computadora.
2. Transferencias DMA.
3. Registros sobre la tarjeta que hacen direccionamiento de los pixeles.
4. Operan modos de detención continua o de fotografía.

5. Su biblioteca provee la selección de una de cuatro fuentes de video. (tres externas y una interna para diagnóstico).
6. Posibilidad de realizar acercamientos.
7. Posibilidad de seleccionar mapas de las LUT's.
8. Bus de video.
9. Sincronización de video americano (60 Hz, 525 líneas) o europeo (50 Hz, 625 líneas).
10. Video activo: 51.2 μ s.
11. Resolución de 512 x 512 o 640 x 512 pixeles, dependiendo del modelo.
12. 8 bits por pixel.
13. Las imágenes que estén en el *buffer* de imagen de la tarjeta pueden ser almacenadas en disco o en la memoria de la PC.
14. Cuatro LUT's, una de entrada y tres de salida, cada una con ocho planos a escoger.
15. Uno o más espacios de trabajo, según el modelo.
16. Cuenta con una biblioteca que proporciona control sobre el hardware. Esta biblioteca es compatible directamente con Microsoft C y FORTRAN.
17. Soporte de operaciones de procesamiento de imágenes.
 - 17.1. Control del hardware (lutm, mask, mode, offset, quadm, sync, video).
 - 17.2. Entrada/salida (colr, colw, rowr, rowl, pixr, pixw).
 - 17.3. Manipulación de LUT (dhisto, histo, luts, scaling).
 - 17.4. Rutinas de dibujo (border, circf, elip, grid, moveto, rect, text, tfont).
 - 17.5. Rutinas de imageonología (dilata, erode, fil3, erode, map, sh1, sh2, con3).

II.1.2. Software

Originalmente el sistema BIOCUM fue escrito en lenguaje FORTRAN, debido a que dicho lenguaje fue ingeniado para aprovechar mejor las capacidades de las computadoras para hacer cálculos numéricos y además era, y sigue siendo, el de mayor difusión dentro de la comunidad científica debido a su amplio repertorio de código fuente desarrollado para dicho lenguaje; sin embargo, dentro del programa también se pueden hallar algunas rutinas en lenguaje C.

Producto de la necesidad de usar ambos lenguajes, surgió la urgencia de poseer una forma en que los compiladores de dichos lenguajes generasen código objeto que permitiera llamadas comunes entre ellos y obtener código ejecutable conteniendo rutinas hechas en ambos lenguajes, aprovechando lo mejor de ambos mundos. Para hacer la interfaz entre ambos lenguajes se cuenta con la ayuda de los compiladores C y FORTRAN de la empresa Microsoft(8,9,10). En ellos se puede realizar la interfaz requerida entre ambos lenguajes de manera sencilla por medio de estructuras o palabras reservadas que contiene como parte de su desarrollo, (ver apéndice F).

Con estas convenciones es posible ligar los códigos objeto generados por ambos compiladores y obtener un código ejecutable completamente compatible, dejando de lado las

distintas formas en que ambos compiladores manejan las llamadas a subrutinas y el paso de sus argumentos.

También es necesario hacer referencia a variables de bloques comunes en FORTRAN, que son áreas de memoria que pueden ser referenciadas por cualquier unidad en un programa y que permiten incrementar el rendimiento eliminando el sobre-encabezado por el paso de argumentos; y a variables globales en C.

Para la implementación en C se crean estructuras globales del tipo externo que contienen a cada variable del tipo común en FORTRAN a las que se hará referencia. Dentro de la estructura en C se debe redeclarar el tipo y tamaño de cada variable de tipo común en FORTRAN. Un buen ejemplo de esto se encuentra en la implementación de las variables BAN1, BAN2 y AC (que serán explicadas en el capítulo cuatro) dentro del archivo biaux.h y que al mismo tiempo son variables comunes en FORTRAN.

II.1.3. Manejo de memoria

Como se mencionó al principio BIOCOM está funcionando sobre maquinas PC así que de alguna forma se debe tener muy en cuenta el tamaño del código fuente; es decir la famosa barrera de los segmentos de 64 kbytes con el que originalmente nacieron las máquinas IBM PC compatibles. Este problema se ha resuelto por medio del uso de la técnica de Overlay que, básicamente, permite que solo se encuentre en memoria el módulo del programa que es necesario en cierto momento de ejecución. Usando esta técnica, y las ventajas de una programación semiestructurada, se dividió al programa en módulos que pueden funcionar de manera independiente o por lo menos semidependiente. El esquema del programa se puede ilustrar como se ve en la siguiente figura:



Fig 2.2 Estructura lógica de cada uno de los módulos que forman BIOCROM.

donde cada módulo será un bloque de código ejecutable que será puesto en memoria, es decir un segmento *overlay*. En realidad, los módulos comparten ciertas rutinas, constantes y variables lo que hace que no sea precisamente cada módulo el que representa un segmento *overlay*; empero, para el entendimiento del programa se puede considerar que cada módulo será visto de esa manera.

II.1.4. Estructura

En BIOCROM existe un módulo decodificador de comandos que además sirve como una especie de controlador del programa. Es decir que este módulo controla el acceso a cada rutina encendiendo o apagando banderas y llamando a las rutinas que se necesitan.

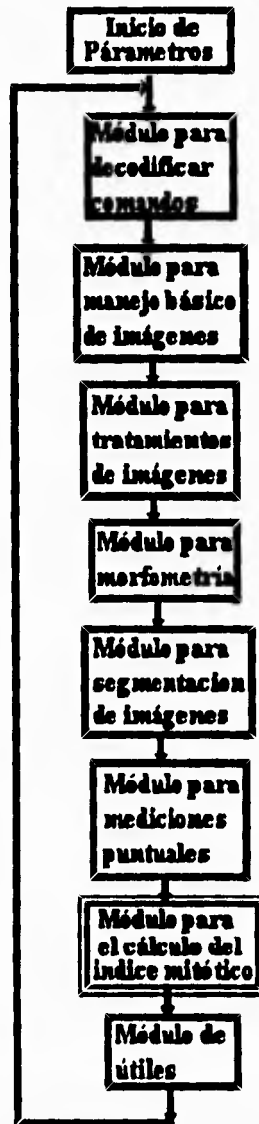


Fig 2.3 Estructura del sistema BIOCUM al agregar un nuevo módulo.

La forma de ciclo en el programa es importante pues permite que se ejecuten comandos de más de un módulo recorriendo cada uno de ellos, como se puede ver en la figura 2.3. Bajo esta filosofía se implantó en el programa la opción de ejecución de archivos tipo texto que contienen comandos del programa BIOCUM; dicha operación se lleva a cabo gracias a al módulo decodificador y a las banderas que maneja. Básicamente se usa un *buffer*, el cual guarda el nombre del comando a ejecutar. Si el comando no fue ejecutado al regresar al módulo decodificador entonces se genera un error de comando equivocado.

Tomando en cuenta la estructura y los beneficios del sistema BIOCUM se decidió que la biblioteca para el cálculo del índice mitótico pasará a ser un módulo más en la estructura del programa. Con esta nueva implementación de la biblioteca para el cálculo del

índice mitótico, la estructura del programa BIOCUM 2000 quedo como se muestra en la figura 2.3.

Contando con las ventajas que ofrecen la plataforma BIOCUM y la tarjeta MATROX la implantacion de la biblioteca para el cálculo del índice mitótico cuenta con una base sólida, las herramientas están disponibles. El siguiente paso es demostrar que dicha biblioteca es realizable.

CAPITULO III

INTERFAZ CON EL MICROSCOPIO

Para llevar a cabo el proyecto es necesario que la computadora pueda obtener información de los cultivos celulares medicados. La computadora necesita tener un acceso ininterrumpido a esta información sin que el usuario este proporcionándola de forma continua. Para ello, la computadora necesita, obviamente, control sobre las fuentes de información. La figura 3.1 muestra la conexión de la computadora y sus periféricos.

III.1. Conexión física

En la primera parte del trabajo se mencionó que la tarea de muestreo y conteo normalmente se realiza bajo el microscopio óptico, esto no cambia en este sistema; sin embargo, la computadora requiere acceso al microscopio óptico, que se logra por medio del uso de una cámara de video **COHU** que trabaja en niveles de grises, video NTSC y **RGB**, conectada al microscopio mediante de una junta **C**, que adquiere las imágenes y las transfiere a la tarjeta **MATROX**, que como ya se dijo tiene entradas para video.



Fig. 3.1 Esquema de la conexión microscopio, cámara y PC

III.2. Controlador del movimiento

Mantener el uso del microscopio como fuente de información brinda la ventaja de poder monitorear el funcionamiento del sistema durante el trabajo del mismo. Sin embargo, el uso del microscopio también representa otra dificultad y es la de tener control sobre las platinas del microscopio y el enfoque del mismo.

III.2.1. Hardware

Para resolver el problema de las platinas y las perillas de enfoque se cuenta con un controlador **MultiControl 2000** de motores de paso. El **MultiControl 2000** es manufacturado por **MÄRZHÄUSER GmbH Wetzlar**. Este controlador tiene tres intérpretes que soportan los comandos del lenguaje interfaz **VENUS-1(15)**.

Para poder controlar las funciones de los motores de paso el controlador cuenta con un CPU RISC de 32 bits, el cual otorga la habilidad al **MC2000** de hacer muchas tareas en tiempo real. Algunas especificaciones del **MultiControl 2000** son:

- Soporta el manejo de dos o tres ejes cartesianos.
- Posee una resolución de 40,000 pasos/revolución.
- Una velocidad máxima de 25 revoluciones/seg.
- Velocidad máxima de 100 mm/seg en estado de pitch de 4mm.
- Velocidad mínima de 0.1 $\mu\text{m}/\text{seg}$ en estado de pitch de 4mm.
- Aceleración máxima de 1000 mm/seg² en estado de pitch de 4mm.
- Velocidad máxima del joystick de 25 rev/seg.
- Máximo consumo de energía 40 Watts.
- Uso de un algoritmo especial para reducir el ruido de movimiento.
- Ajuste dinámico de corriente para los diferentes motores de pasos.
- Interfaz física : RS232.
Enlace de transputer.
IEEE 488 opcional.
SCSI-2 opcional.
- Intefaz por software: Emulador MCL.
Emulador MAC 4000.
VENUS-1
- Sistema de coordenadas programable.
- Velocidad y aceleración programable.
- Aprendizaje de los límites de estado y límites programables.
- Memoria no volátil para los parámetros.
- Actualización de software sin cambio físico.
- Uso de unidades reales. El controlador se encarga de transformar los movimientos de rotación a su equivalente lineal.

III.2.2. Software

Como se puede observar se poseen varias ventajas al usar este controlador; empero, una de las más importantes es la del lenguaje **VENUS-1** que es un lenguaje orientado a pilas como el **Postscript**. El **MultiControl 2000** brinda grandes facilidades de programación en su modo *huésped* por medio del lenguaje **VENUS-1 (15)**.

III.2. Controlador del movimiento

Mantener el uso del microscopio como fuente de información brinda la ventaja de poder monitorear el funcionamiento del sistema durante el trabajo del mismo. Sin embargo, el uso del microscopio también representa otra dificultad y es la de tener control sobre las platinas del microscopio y el enfoque del mismo.

III.2.1. Hardware

Para resolver el problema de las platinas y las perillas de enfoque se cuenta con un controlador **MultiControl 2000** de motores de paso. El **MultiControl 2000** es manufacturado por **MÄRZHÄUSER GmbH Wetzlar**. Este controlador tiene tres intérpretes que soportan los comandos del lenguaje interfaz **VENUS-1(15)**.

Para poder controlar las funciones de los motores de paso el controlador cuenta con un CPU RISC de 32 bits, el cual otorga la habilidad al **MC2000** de hacer muchas tareas en tiempo real. Algunas especificaciones del **MultiControl 2000** son:

- Soporta el manejo de dos o tres ejes cartesianos.
- Posee una resolución de 40,000 pasos/revolución.
- Una velocidad máxima de 25 revoluciones/seg.
- Velocidad máxima de 100 mm/seg en estado de pitch de 4mm.
- Velocidad mínima de 0.1 $\mu\text{m}/\text{seg}$ en estado de pitch de 4mm.
- Aceleración máxima de 1000 mm/seg² en estado de pitch de 4mm.
- Velocidad máxima del joystick de 25 rev/seg.
- Máximo consumo de energía 40 Watts.
- Uso de un algoritmo especial para reducir el ruido de movimiento.
- Ajuste dinámico de corriente para los diferentes motores de pasos.
- Interfaz física : RS232.
Enlace de transputer.
IEEE 488 opcional.
SCSI-2 opcional.
- Intefaz por software: Emulador MCL.
Emulador MAC 4000.
VENUS-1
- Sistema de coordenadas programable.
- Velocidad y aceleración programable.
- Aprendizaje de los límites de estado y límites programables.
- Memoria no volátil para los parámetros.
- Actualización de software sin cambio físico.
- Uso de unidades reales. El controlador se encarga de transformar los movimientos de rotación a su equivalente lineal.

III.2.2. Software

Como se puede observar se poseen varias ventajas al usar este controlador; empero, una de las más importantes es la del lenguaje **VENUS-1** que es un lenguaje orientado a pilas como el Postcript. El **MultiControl 2000** brinda grandes facilidades de programación en su modo *huésped* por medio del lenguaje **VENUS-1 (15)**.

Los comandos del lenguaje VENUS-1 pueden dividirse en comandos de servicio, de configuración, de encendido, movimiento, despliegue, interactivos, de control y del manejo de la pila (*stack*).

Algunos de los comandos que se usaron son:

Nombre	Prototipo	Función
selftest	n selftest	Verifica el estado del controlador al encendido
setaxis	i n setaxis	Habilita o deshabilita el eje n
joy	n j	Habilita o deshabilita el uso del joystick
cal	cal	Mueve las platinas al límite inferior en x e y. Debe ser ordenado antes de cualquier otra acción de movimiento o medición.
rm	rm	Mide los límites mayores de las platinas. Se manda después de <i>cal</i> .
mode	n mode	Pone en modo terminal o host al controlador
setdim	n setdim	Activa las dimensiones del marco de referencia
move	x y z m	Movimiento absoluto a x,y y z
rmove	x y z r	Movimiento relativo a x,y y z
pos	p	Obtiene la posición de las platinas
status	st	Obtiene el estatus de las platinas
setunit	i n setunit	Pone las unidades de medida para el eje n
setvel	v setvel	Aplica la velocidad v al movimiento de las platinas
setjoyspeed	v setjoyspeed	Aplica la velocidad v al movimiento que provoca el joystick
setpos	0 0 0 setpos	Actualiza el origen de las coordenadas

Para que el sistema sea completamente automatizado se requiere de tener el control sobre los movimientos de las platinas y la obtención de imágenes. Con las rutinas de BIOCOM, de la tarjeta MATROX y los comandos del MultiControl 2000 se pueden realizar

las rutinas para el enfoque en cada campo, posición en un campo deseado y barridos de los portaobjetos.

III.3.1. Enfoque

III.3.1. Contraste

El enfoque de imágenes está muy relacionado a la afinación de ellas. Para lograr el enfoque de una imagen es necesario que la computadora tenga cierto control sobre el microscopio o el lente de la cámara.

Un método para enfocar una imagen requiere de hacer pasar la señal de video a través de un filtro que deja pasar las frecuencias altas, paso-altas. Cuando se encuentra la máxima fuerza de una señal a altas frecuencias entonces se halla el punto de mayor afinamiento y por lo tanto el punto de mejor foco (8).

Algunos métodos para encontrar el valor máximo de una imagen afinada son:

$$F_1 = \sum_i \sum_j (g_{i,j} - g_{i+1,j})^2$$

$$F_2 = \sum_i \sum_j g_{i,j}^2 - \sum_i \sum_j g_{i,j} \cdot g_{i+1,j}$$

$$F_3 = \sum_i \sum_j (g_{i,j} - \mu)^2 \quad \text{donde } \mu \text{ es la media de brillantez en la región de muestra}$$

$$F_4 = \sum_i \sum_j g_{i,j} \cdot g_{i+1,j} - \sum_i \sum_j g_{i,j} \cdot g_{i+2,j}$$

$$F_5 = \sum_i \sum_j g_{i,j} \cdot g_{i+1,j} - N \cdot M \cdot \mu^2 \quad \text{para una subregion rectangular de tamaño } M \times N$$

Para realizar sumas sobre la imagen o segmento de la imagen se usa g que representa el valor de la brillantez, en la escala de grises, de cada pixel en la posición i,j . Hacer máximo el valor de F en cualquiera de las fórmulas de arriba, producirá un punto de enfoque óptimo.

Como se mencionó anteriormente es posible lograr el enfoque de un dispositivo óptico por medio de un algoritmo que calcule el contraste en una serie de tomas y eligiendo como la de foco óptimo a la que posea el mayor contraste.

El algoritmo para obtener el contraste de una toma que se usa en el sistema es el que se esquematiza en la figura 3.2 y se resume como sigue:

Encontrar la diferencia de los pixeles claros y oscuros de la pantalla varias veces y seleccionar el valor que sea mayor.

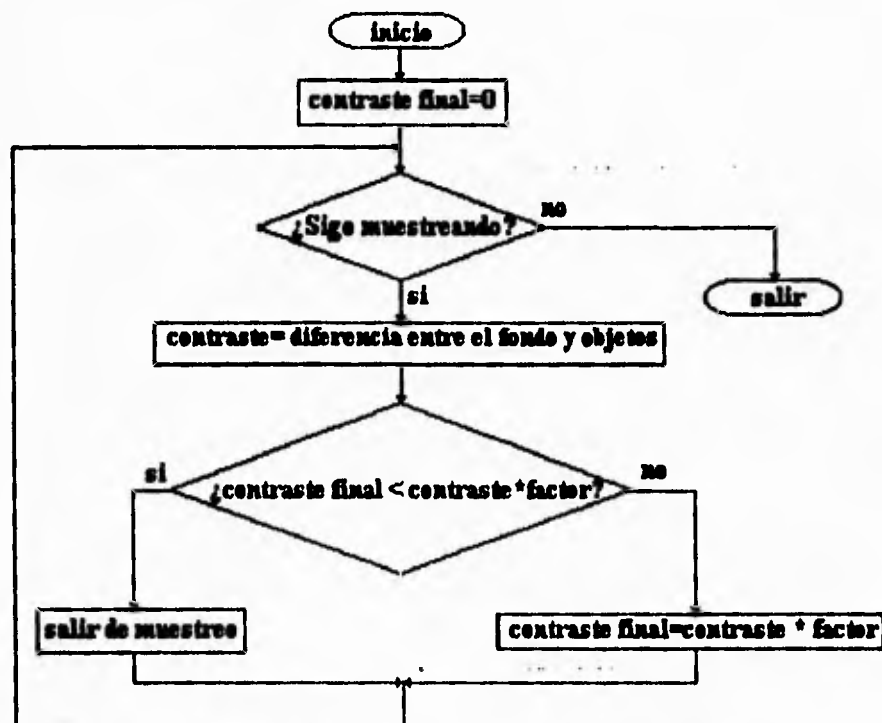


Fig 3.2 Esquema para hallar el contraste de una imagen.

El factor que multiplica a la diferencia de cada toma es un valor que se relaciona con el número de veces que se hace el muestreo y el número de líneas que se usan para el muestreo. La implementación del algoritmo anterior puede ser visto en la rutina `ini_cont1`.

III.3.2. Movimiento

Para realizar el enfoque en una posición (x,y) específica del controlador, y de la muestra, es necesario realizar un barrido en el eje z (eje de enfoque) que abarque una distancia lo suficientemente grande para que el proceso siempre encuentre el foco de la toma.

Una de las opciones era implementar un algoritmo similar al método de divisiones consecutivas de Newton, esta opción permitiría tener un sistema de enfoque que pudiera barrer casi todo el espectro en el eje z. Sin embargo esta alternativa oscilaba mucho, haciendo que el enfoque fuera muy lento. Los problemas se debieron a que es necesario poner bien la semilla de arranque y también a que no siempre cada posición en z tiene el mismo nivel de contraste; este efecto es producido porque la cámara de video tiene un

margen de ganancia y al igual que la tarjeta, realiza un muestreo que ocasiona una pequeña variación de contraste.

Finalmente se decidió implementar un enfoque limitado, es decir que el enfoque barrera solo parte del espectro en z, hallando el foco de la posición (x,y) entre dichos límites. El enfoque barre, desde el punto original, seis posiciones anteriores y seis posteriores sobre el eje z, como puede observarse en la figura 3.3. Este límite de pasos puede cambiar siempre que no se halle un punto óptimo o sea que todas las muestras tengan el mismo nivel de contraste.

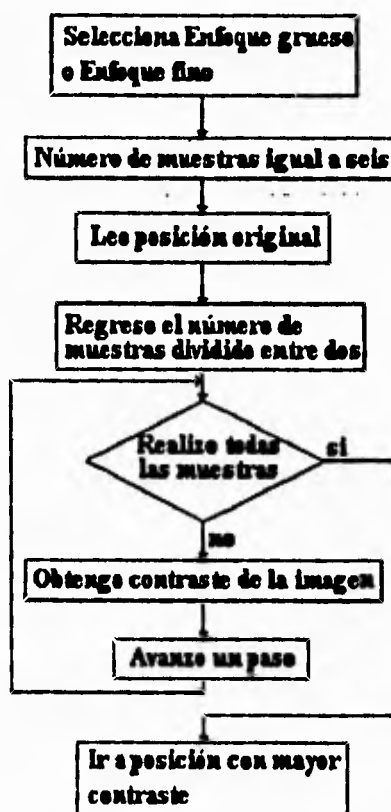


Fig. 3.3 Esquema de la rutina de enfoque.

La dimensión del paso depende del tipo de enfoque que se desee pues el sistema permite el enfoque burdo (grueso) o fino, el algoritmo no cambia solo cambia el tamaño del incremento.

Al algoritmo se le han hecho algunas modificaciones en su implementación en la rutina **enfoco**, como el definir un límite; esto es para evitar que al faltar la energía eléctrica que alimenta la iluminación del microscopio el sistema llegue a romper la laminilla que se encuentra examinando. Este límite se definió de manera experimental y depende del tipo de objetivo que se esté usando para la revisión del portaobjeto.

De manera similar se cuenta con una variable para saber la posición actual del sistema sobre el eje Z (off_z1). El uso de esta variable se debió a problemas con el controlador MC2000 ya que después de demasiados comandos, el controlador se quedaba en un estado suspendido provocando que el sistema dejara de funcionar en su totalidad.

III.4. BARRIDOS

III.4.1. Movimiento

Se mencionaron las facilidades del controlador **MultiControl 2000** pues posee comandos de movimiento en los tres ejes coordenados, empero, es necesario realizar una implementación de una rutina que mande los comandos al controlador; se requiere también que dicha rutina espere respuesta del mismo controlador con referencia a que los motores han llegado a la posición requerida. Esta rutina se llama **muevexyz** que permite el movimiento de forma absoluta y relativa al punto en que se halle el controlador. Para hacer la espera de la terminación del movimiento se obtiene el estatus del controlador hasta que permite ejecutar otro comando.

Por lo mencionado anteriormente, puede decirse que ya se posee control sobre la fuente de información de los cultivos. Sin embargo, como se mencionó en la primera parte del trabajo, existe un barrido específico de las laminillas así que es indispensable emular dicho barrido.

III.4.2. Desplazamientos

Antes que nada es necesario identificar el área que es visible en cada pantalla usando un objetivo determinado, es decir al usar un objetivo del microscopio con un lente de n aumentos se puede apreciar un plano (x,y) que difiere del observado bajo un objetivo de otro aumento. La información sobre el área que abarca cada objetivo es guardada en un archivo de definiciones llamado **OBJETIV.MIC**.

III.4.3. Definición de Areas de Observación

El sistema cuenta con la facilidad de poder cambiar de campo de definición o de redefinir el área de un objetivo. La definición del área para cada objetivo se realiza de la siguiente manera: se pintan en la pantalla del monitor "imagen" cuatro cruces que servirán como referencia, se pide al usuario que mueva un objeto a cada una de las cruces, validando su llegada a cada una de ellas, y una vez en cada cruz se lee su posición en (x, y) para posteriormente hacer los cálculos del área demarcada (ver figura 3.4).

La implementación de esta idea puede verse en la rutina llamada **defobj**. En esta rutina se encontrará una idea más que permite verificar si el movimiento en x e y es paralelo a los ejes x-y.

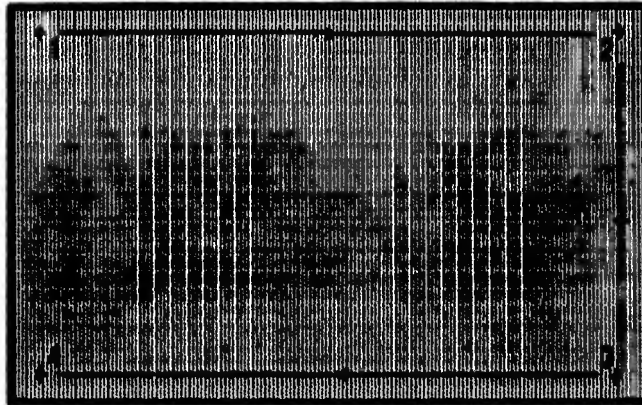


Fig 3.4 Pantalla de movimiento grafico

III.4.4. Selección del Area de Observación

La elección de un objetivo es un poco más simple pues se lee el archivo **OBJETIV.MIC**, en donde están todos los objetivos ya definidos y el objetivo que se encuentra activo. Para cambiarlo se realizó una interfaz gráfica para poder elegir por medio del ratón el objetivo deseado. La idea es leer el contenido del archivo e imprimirlo en la pantalla. Con ayuda de la rutina para el movimiento del ratón es posible leer en que posición de la pantalla se encuentra y, por medio de una conversión de coordenadas de la pantalla a la posición del archivo, es posible saber que objetivo se está seleccionando. Esta operación se lleva a cabo por la subrutina **actobj**.

Tipo	Despx	Despy	Activo
x4	5526.28125	1076.05078	
x10	2208.00000	435.82422	*
x20	1104.62500	216.72266	

Fig 3.5 Pantalla de selección de objetivos

III.4.5. Traslape

Cabe mencionar que el recorrido de las tomas no está ajustado pues se realiza un pequeño traslape de ellas para que posibles objetos que hubiesen quedado en medio de cada toma puedan ser revisados. Este traslape es definido por el usuario para cada tipo de objetivo (función **encima**) y es almacenado en el archivo **OVERLAP.PAR** y en la variable **overlap**. Este traslape normalmente es del tamaño promedio del objeto más grande a reconocer pero es responsabilidad del usuario definir bien los traslapes para que no queden cuerpos sin contar ni se cuenten cuerpos de más.

III.4.6. Límites

Hablando de los límites del controlador, éste tiene definido el valor mínimo y máximo del eje X y es imposible modificarlo así que el valor del tamaño en X de las laminillas es más o menos constante; sin embargo, el valor en Y es modificable así que es preciso que al calibrar las platinas se lea el tamaño del desplazamiento total sobre el eje Y. Esto se hace normalmente al entrar al programa (función **inicia**).

III.4.7. Zona Valida de Barrido

Como las muestras de cultivo normalmente ocupan una sección de la laminilla, es inútil barrer toda el área de la laminilla; para esto se define un porcentaje de las laminillas que será desechado del barrido, como puede verse en la figura siguiente.

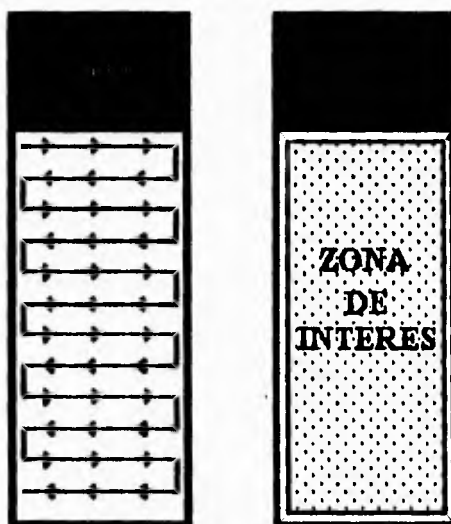


Fig. 3.6 Forma del barrido y zona de interés en la laminilla.

Las rutinas que definen estos bordes finales son **despx** y **por100**. Al igual que la definición del área para cada objetivo los últimos límites de barrido para cada laminilla son almacenados en un archivo **DESP.PAR**.

III.4.8. Tipos de Barrido

El área definida anteriormente es la de un barrido predefinido, sin embargo, es posible que el usuario redefina la zona de interés para todas las laminillas o para cada laminilla instalada. Esto se logra por medio de las rutinas **defind** y **deflx8** respectivamente. Es importante mencionar que la forma en que se realiza el barrido nunca cambia pero si es posible cambiar la zona de barrido. Las zonas de barrido para cada laminilla son almacenadas en la variable **region**.

Cada vez que se realiza una ejecución de barrido se calcula el número de tomas que se harán por cada eje; esto lo realiza la función **calcula_mov** que realmente hace un calculo de la relación, para cada portaobjetos, del tamaño de los límites definidos en **region** y el tamaño del área que abarca el objetivo (**factor**), guardando dicha relación en la variable **veces**. Es por ello que antes de realizar alguna orden de barrido se elija el objetivo correcto y se defina la zona de barrido.

Por último debe mencionarse que después de cada localización en x,y es indispensable enfocar el campo; posteriormente se lleva a cabo la operación del

reconocimiento de los cuerpos en dicho campo. Este esquema puede verse en la siguiente figura.

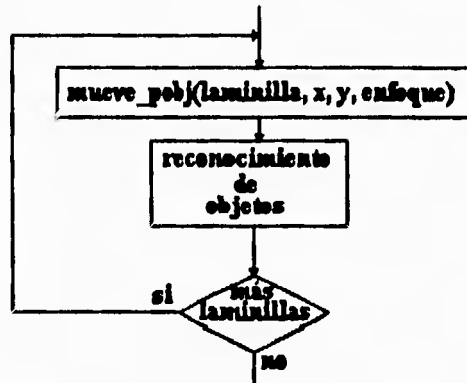


Fig. 3.7 Funcionamiento básico del funcionamiento del sistema.

Es necesario tener un control sobre la posición en que se encuentra el barrido para cada laminilla, este control se lleva a cabo por medio de la variable veces y dentro de la función mueve_pobj. Esta función realiza el control del sentido del barrido, el cambio de campo, enfoque de la toma y cambio de portaobjetos. A continuación se muestra un esquema del funcionamiento de mueve_pobj.

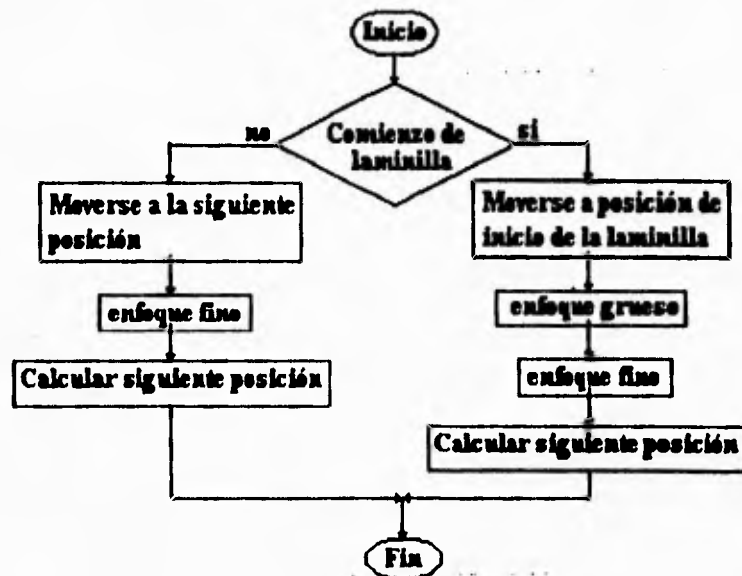


Fig 3.8 Forma en que se realizan la operación de control dentro de mueve_pobj.

Como se puede ver, el enfoque grueso solo se realiza al comienzo de cada laminilla y en el resto del portaobjetos se realiza un enfoque fino.

III.4.9. Manejo de Controles y Experimentos.

Para hablar de la forma en que se hizo la implementación final para calcular el índice mitótico siguiendo las especificaciones de los biólogos, es necesario mencionar las indicaciones que se deben seguir para revisar las laminillas. Se ha mencionado el uso de controles y laminillas asociadas a dichos controles; esto se debe, como se recordará de la introducción, a que si el cultivo es muy grande se puede tener un número alto de laminillas que constituyen el experimento, así que se toma una muestra de laminillas que serán revisadas primero (llamadas controles). Si estas laminillas cumplen un índice mitótico mínimo, entonces se revisa el resto de las laminillas que conforman el experimento (a éstas se les llama experimentos asociados).

Esta forma de revisión debía ser implementada en el sistema; así que debía desarrollarse la forma en que se pudiera detectar que posición de las platinas pertenecía a qué categoría.

Se desarrolló el siguiente sistema: se tienen un número fijo de portaobjetos instalados, definido en la variable `num_portaobj`, y un número fijo de posibles posiciones (el sistema de platinas motorizadas puede soportar como máximo ocho laminillas). Se implementaron las siguientes estructuras para poder controlar el manejo de las laminillas: `controles[8]` y `asociados[8]`, ambos del tipo carácter. En la estructura `controles` se sabe que tipo de laminilla ocupa cada bahía de las laminillas; las laminillas que representan a los controles se representan con las `c` mayúscula y minúscula. Las laminillas que representan a los experimentos se representan con las letras `e` mayúscula y minúscula y, por último, las laminillas que no dependen de ninguna otra laminilla se denominan libres y son representadas por las letras `l` mayúscula y minúscula. Cuando una laminilla no ha sido revisada aún, siempre habrá letras mayúsculas en cada posición de la estructura `controles`, una vez que fueron revisadas la letra cambia a minúscula.

La estructura `asociados` funciona de la siguiente manera: el casillero que representa a un control es el que se modifica los demás tendrán el valor de cero. Si un control tiene experimentos asociados entonces estos serán representados por medio del encendido del bit que refleja su posición dentro de las platinas. Si la laminilla 1 es un control y tiene asociados a las laminillas 4 y 7 entonces las estructuras quedan como sigue:

- `controles[0]='C'`, `controles[3]='E'`, `controles[6]='E'`.
- `asociados[0]=0+0+0+8+0+0+64+0=72`.

Como es posible observar, la posición de cada laminilla es menor por una unidad debido a como se hace referencia a arreglos en el lenguaje C.

Antes de iniciar el cálculo del índice mitótico se renuevan las estructuras y se pide el tipo de laminillas que se encuentran instaladas. La rutina que se encarga de realizar esto se llama **asocia**.

Las rutinas **findnextcontrol**, **findnextlibre** y **findnextexpasoc** son otras rutinas que modifican las estructuras **controles** y **asociados**. La rutina **findnextcontrol** hace una consulta de la estructura **controles** para hallar si hay la posición de la siguiente laminilla del tipo control, si ya no hay laminillas control entonces regresa el valor de cero. Si, por el contrario, hay una casilla que contenga la letra C mayúscula, entonces la cambia por c minúscula y regresa la posición en que la encuentro.

La rutina **findnextlibre** funciona de la misma forma que la rutina **findnextcontrol**, solo que con las letras *l* minúscula y L mayúscula.

En el caso de la rutina **findnextexpasoc** el acceso se hace a la estructura **asociados** por medio de una variable que indica el control que se usa en ese momento, llamada **control_ant**. Si la estructura **asociados[control_ant]** tiene algún valor distinto a cero, entonces se realiza una máscara del primer bit que este prendido, desde el menos significativo hasta el más significativo, que representara el siguiente experimento a revisar. Dicho bit es apagado después de ser hallado. Cuando no hay más experimentos asociados para **control_ant** entonces la rutina regresa un valor negativo.

También se hace uso de la variable **ant** que acarrea el valor del último portaobjetos que fue revisado.

CAPITULO IV

CALCULO DEL INDICE MITOTICO.

IV.1. Categorías

Al comienzo del trabajo se hizo una introducción acerca de la forma en que se realiza el cálculo del índice mitótico reconociendo dos categorías. La primera es la de los núcleos estimulados y la segunda la de las mitosis. Sin embargo, en las preparaciones existen núcleos no estimulados y artefactos. La diferencia física entre los núcleos estimulados y los no estimulados es que los primeros son más grandes y claros que los segundos. En cuanto a las mitosis, existen de dos tipos compactas y dispersas cuya diferencia radica, básicamente, en el área que ocupan en la pantalla.

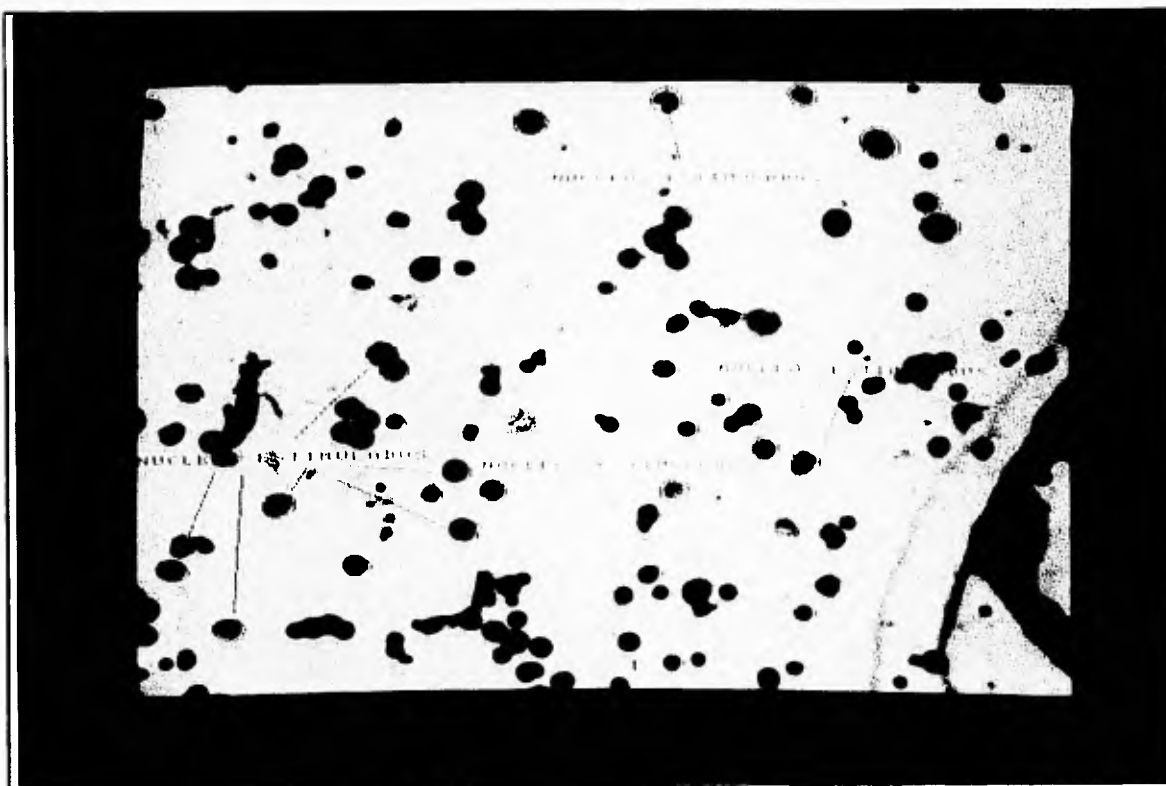
IV.1.1. Núcleos

Los núcleos son los cuerpos que se presentan en mayor cantidad en las preparaciones; cuando los núcleos se clasifican manualmente presentan un tamaño en cierto rango, una forma casi circular y un color característico lo que los hace fácilmente identificables al realizar el cálculo manualmente, ver ilustraciones de la figura siguiente.



a)

Fig. 4.1 Núcleos estimulados. a) ejemplo 1. b) ejemplo 2.



b)

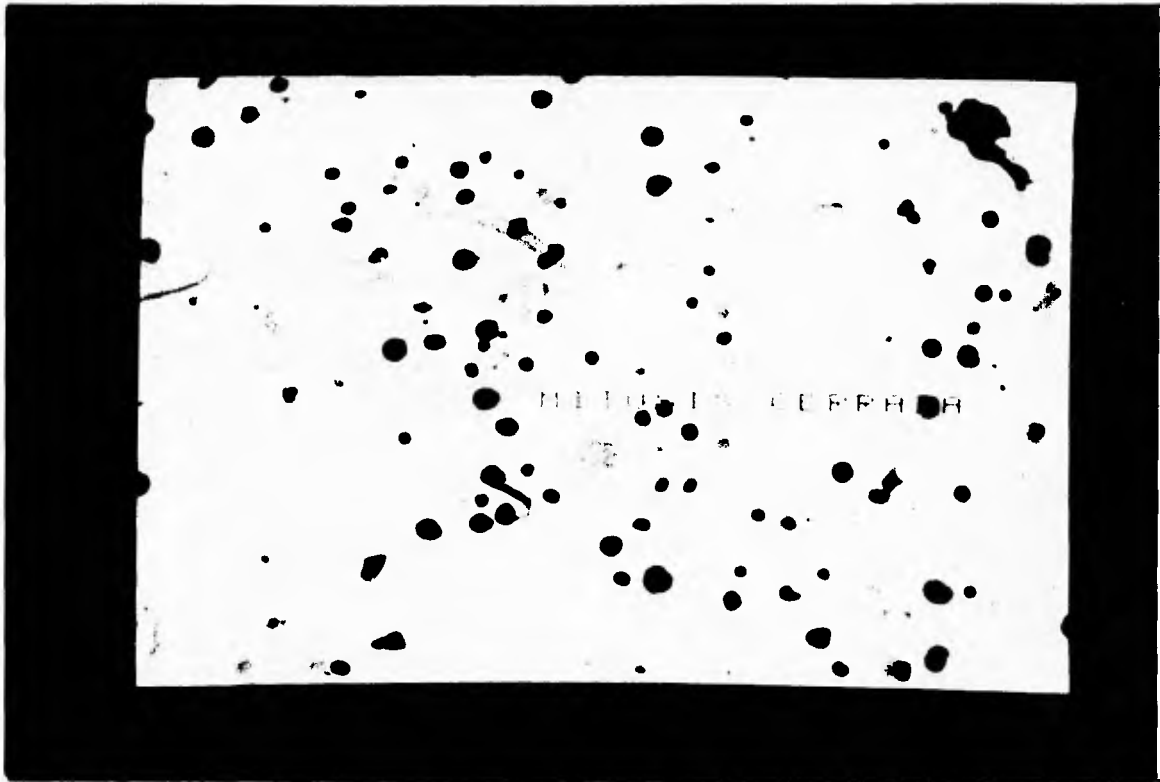
Fig. 4.1 Núcleos estimulados. a) ejemplo 1. b) ejemplo 2.

Sin embargo, es necesario tener cierta experiencia para reconocer a los núcleos estimulados de los no estimulados, pues si en una laminilla predominan los núcleos estimulados grandes entonces el ser humano, por naturaleza, restringe su reconocimiento a dichos núcleos olvidándose de los núcleos estimulados más pequeños, los que quizás están en el límite de estimulados y no estimulados. Para la computadora no existe dicho problema pues siempre contará los núcleos que se le hayan instruido reconocer.

Después de haber sido duplicados en la mitosis, los cromosomas están en proceso de separación y éstos pueden encontrarse en diversas etapas, (figura 4.2) que van desde los que apenas están abandonando el estado de núcleo estimulado hasta los cromosomas completamente esparcidos por el campo de observación. Por esta razón es conveniente, para propósitos del sistema, dividir a las mitosis en dos categorías: mitosis compactas y mitosis dispersas. De aquí en adelante se hablará de ellas por separado pero se debe tomar en cuenta que en la realidad forman parte de una sola clase.

IV.1.2. Mitosis

Las mitosis separadas o dispersas tienen una densidad de cromosomas menor que las mitosis compactas lo que provoca que, en ciertas circunstancias, en las mitosis compactas no se puedan identificar cromosomas individuales sino una aglomeración de objetos, como es posible observar en las siguientes imágenes.

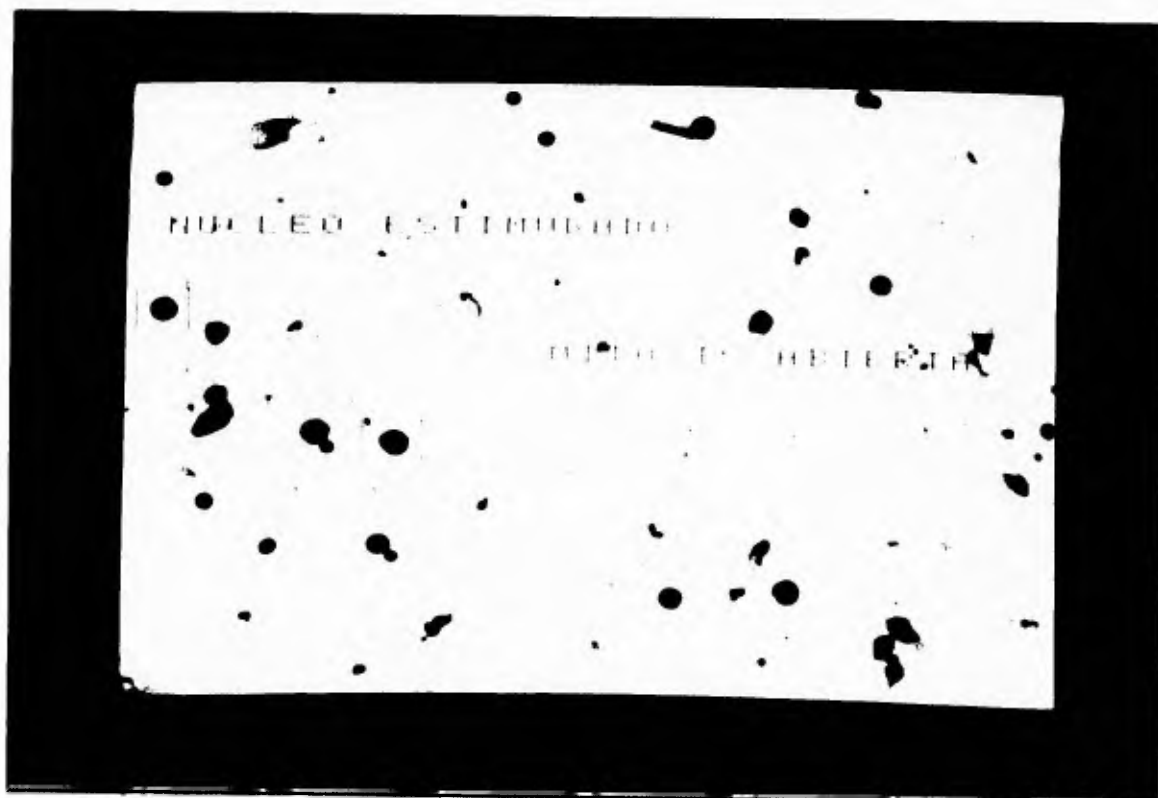


a)

**Fig. 4.2. Mitosis dispersas y compactas. a) ejemplo 1.
b) ejemplo 2. c) ejemplo 3.**

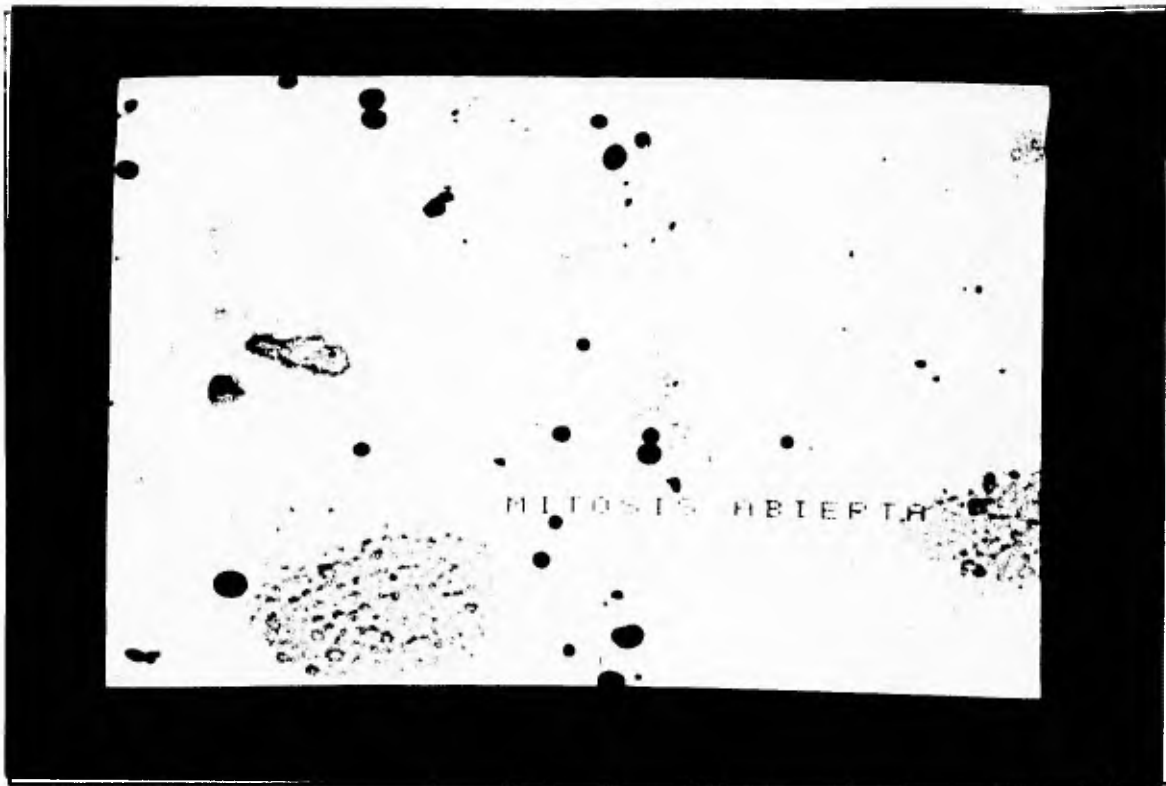


b)



c)

Fig. 4.2. Mitosis dispersas y compactas. a) ejemplo 1.
b) ejemplo 2. c) ejemplo 3.



b)



c)

Fig. 4.2. Mitosis dispersas y compactas. a) ejemplo 1.
b) ejemplo 2. c) ejemplo 3.

En este punto es necesario hacer notar que el ser humano está capacitado para agrupar elementos de un mismo cuerpo y clasificarlos como una sola unidad pero la computadora requiere de una técnica que permita identificar como un solo cuerpo a varios elementos esparcidos en la pantalla para luego obtener sus características. Si se pueden agrupar los componentes de una mitosis entonces el ser humano clasificará a dicho grupo como un núcleo o artefactos dependiendo de, básicamente, su textura.

IV.2. Representación y Descripción de clases

IV.2.1. Segmentación

Antes de comenzar con la definición de parámetros que se usaron para reconocer a los núcleos y a las mitosis, es necesario mencionar que debido a que BIOCOTM posee la capacidad de segmentar la imagen bajo un nivel de gris determinado se necesitaba entonces encontrar un algoritmo que permitiera hallar un nivel de gris que separara el fondo de la imagen de los objetos de interés.

Como se mencionó en el segundo capítulo, Otsu (1) desarrolló un método que divide dos clases de píxeles dentro de la pantalla. Se supuso que una clase es el fondo y la otra los objetos de interés para implementar el algoritmo de Otsu, que encontrará el nivel de gris que define la separación de las dos clases. Esta implementación puede ser revisada en la función `find_umbral` en el disco. La implementación está apegada al método que se menciona en el capítulo 2.

Dicho umbral, o nivel de gris que divide dos categorías, es pasado a las rutinas de BIOCOTM para que segmenten la imagen bajo ese nivel de gris. Esta operación puede ser vista en la función `segmenta` en el disco y su efecto en la figura 4.3. Es necesario tomar en cuenta que al segmentar la imagen esta es congelada.

El método de Otsu realmente permite dividir el fondo de ciertos grupos de píxeles que representan objetos de interés dentro de la pantalla, sin embargo no siempre ocurrió así, a veces se presentaron problemas de saturación o pérdida de información. Los problemas en la segmentación de las imágenes serán tratados en este capítulo pero al momento en que se explique la forma de reconocer las diferentes categorías.

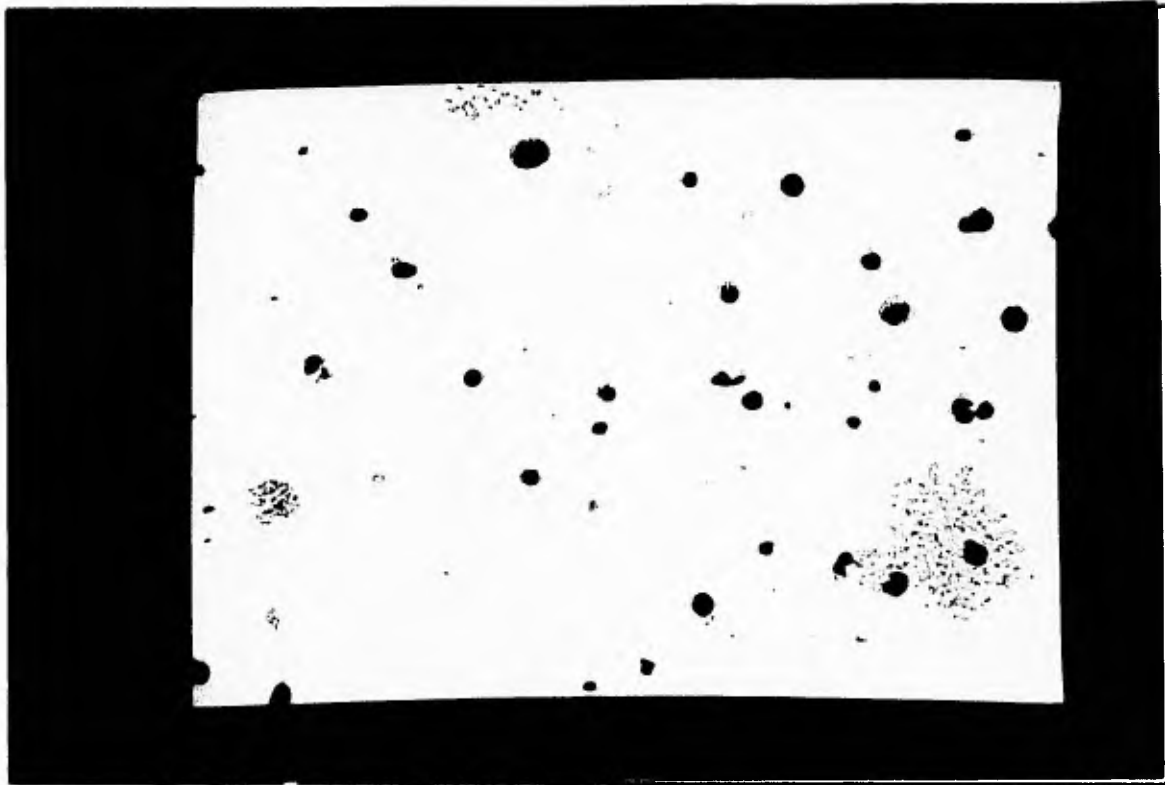


Fig. 4.3. Segmentación de una pantalla.

IV.2.2. Obtención de parámetros

Por otra parte habría que mencionar que una vez hecha la segmentación, entonces se cuentan con objetos, o contornos, que son numerados por el sistema y a los cuales se tiene acceso por medio de tablas existentes en BIOCOT. Sin embargo, para obtener sus características morfométricas no es necesario hacer uso de las tablas, por lo menos no todo el tiempo. Para obtener las medidas morfométricas que BIOCOT permite realizar, se hace uso de la rutina **ROUTI** en FORTRAN, la cual tiene el siguiente prototipo:

ROUTI(tipo, contorno, retorno)

donde *tipo* es el tipo de valor morfométrico que se desea. Para saber qué valor debe tener *tipo* al llamar a **ROUTI** se debe consultar una tabla de BIOCOT o el código fuente de dicha rutina.

contorno es el contorno al cual se le desea sacar la medición.

retorno es el valor obtenido de la medición.

Por medio de esta rutina es posible obtener de un contorno una serie de medidas morfométricas que ya están implementadas en BIOCOT como:

- Coordenada x del centro de gravedad del contorno.
- Coordenada y del centro de gravedad del contorno.
- Area del contorno.

- Perímetro del contorno.
- Factor de forma del contorno.
- Media del nivel de gris dentro del contorno.
- Desviación del nivel de gris dentro del contorno.

IV.3. Búsqueda de parámetros que definen a las distintas categorías

Es importante señalar que mientras se llevaba a cabo el proceso de búsqueda de las características de cada categoría, se realizaron otros procesos que serán tratados después de explicar la definición de las características de cada categoría.

IV.3.1. Núcleos

Para el caso de los núcleos no fue difícil identificar los parámetros que los hacían característicos, simplemente se usaron los mismos parámetros que el ser humano usa para reconocerlos: circularidad, tamaño y color. Esto se traduce a nivel de gris, desviación en el nivel de gris, factor de forma, que es una medida de la circularidad de un objeto, y su área.

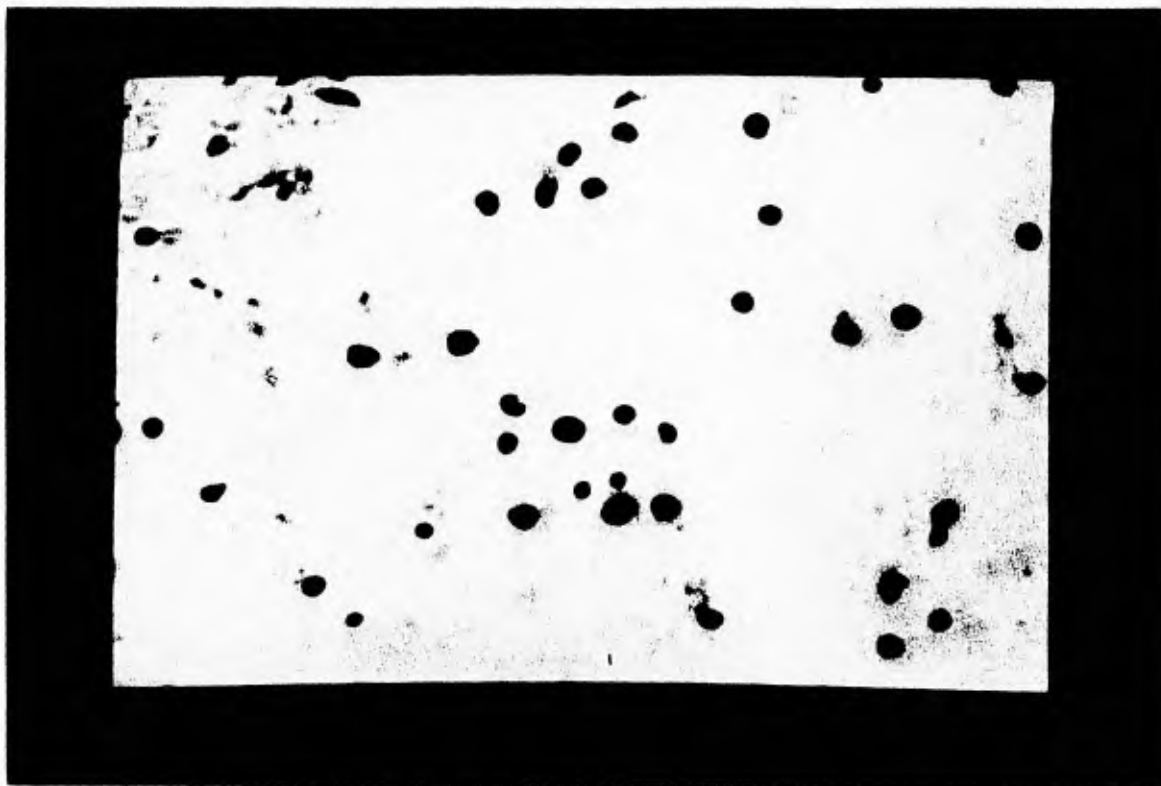
El factor de forma ayudó mucho a discriminar algunos objetos, llamados artefactos, que no son de interés en esta etapa de los que lo son; aunque muchos de los objetos artefactos lograron confundirse como núcleos estimulados. El que el factor de forma sea de gran utilidad se debe a que los núcleos estimulados poseen una forma casi circular, es decir su factor de forma está muy cercano al valor 1.

El parámetro de tamaño al principio fue usado como filtro de dos categorías, núcleos estimulados grandes y núcleos estimulados pequeños. Esto fue debido a que al hacer algunos muestreos manualmente se vio que era posible, también, dividir esta clase en dos categorías sin embargo al realizar las diferentes pruebas del sistema se vio que esta división se podría ahorrar haciendo menos complicado el sistema.

A pesar de usar el tamaño como un buen filtro se seguían presentando problemas con la inclusión de artefactos pues algo que ayuda mucho a los investigadores a discernir entre un núcleo estimulado de uno no estimulado, sobre todo en los límites de las categorías, es el color y la textura de los núcleos así que había que realizar la implementación de este parámetro para lograr un mejor reconocimiento.

El problema surgió en ese momento, debido a que, al tener la imagen directamente del microscopio, ésta se encuentra un poco alisada debido a la forma en que es adquirida la imagen por la cámara y no se posee mucho contraste en cada objeto haciendo difícil desechar objetos con una textura accidentada, que no son núcleos estimulados, y aceptar los de textura más lisa, o núcleos estimulados (ver figura 4.4). Para acentuar el contraste de cada objeto se decidió realizar un proceso a la imagen antes de obtener las medidas de nivel

de gris y desviación estándar del nivel de gris, que sirve como una cierta medida de textura pues da una idea de que tan lejanos están los niveles de grises de un contorno de un valor de gris medio. La medición de estos parámetros no debía modificar las mediciones sobre tamaño y factor de forma, o por lo menos incidir lo menos posible en ellas.



a)

Fig. 4.4. Tipos de Imágenes a) sin contraste. b) contraste.



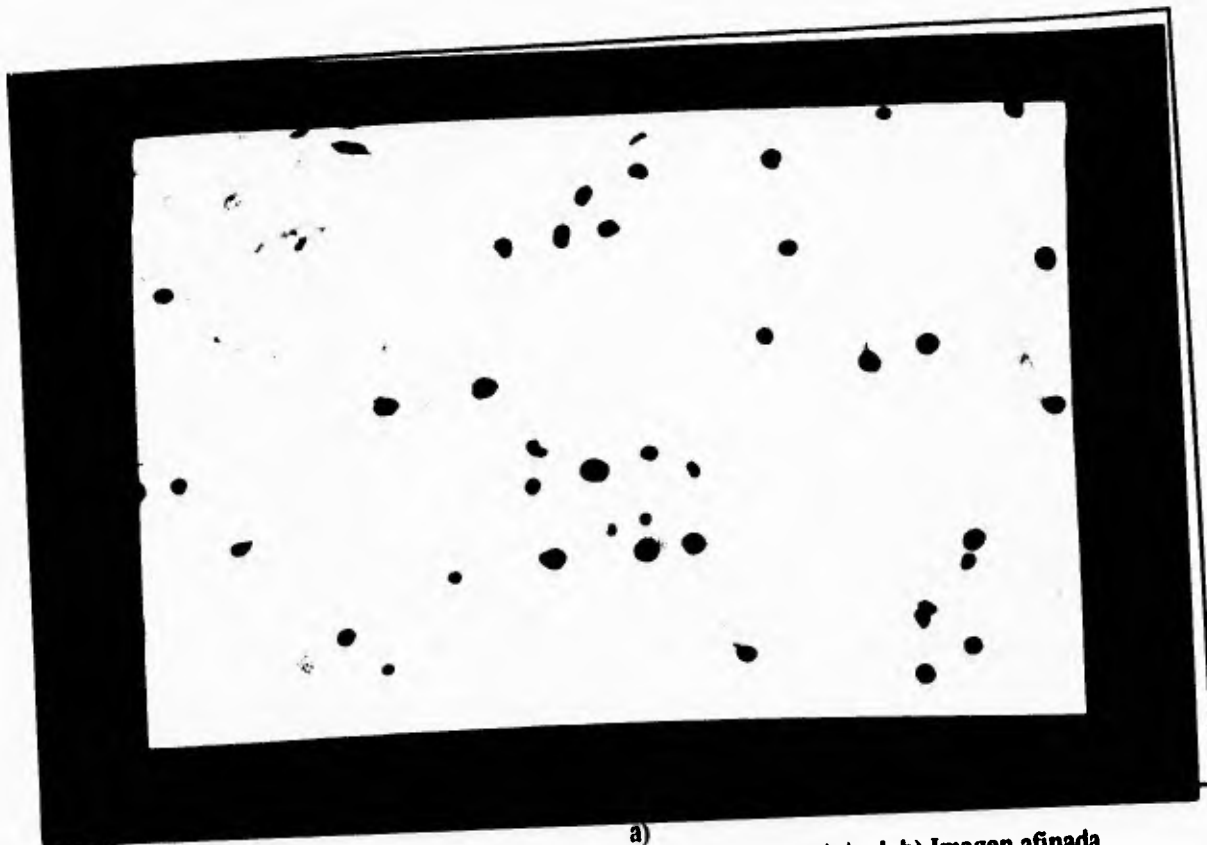
b)

Fig. 4.4. Tipos de Imágenes a) sin contraste. b) contraste.

El primer paso fue afinar la imagen con una máscara del tipo

$$\begin{vmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{vmatrix}$$

que está incluida como una función de la tarjeta MATROX llamada **fg_sh1**. Cuyo resultado puede verse a continuación:



a)

Fig. 4.5. Proceso de afinación de una imagen. a) Imagen original. b) Imagen afinada.



b)

Fig. 4.5. Proceso de afinación de una imagen. a) Imagen original. b) Imagen afinada.

Esta función ayudó bastante, sin embargo se necesitaba de un mayor contraste y definición de los bordes de los objetos, debido a que aún algo de los artefactos se confundían con núcleos estimulados, así que se cambió a la máscara:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

que también está contenida en la tarjeta MATROX, bajo el nombre de **fg_sh2**, y cuyo resultado puede verse en la imagen 4.6. Con esta máscara se obtuvo un mayor contraste haciendo posible discernir entre artefactos y objetos que representan a los núcleos estimulados.



Fig. 4.6. Imagen afinada por segundo método.

Realmente parecía que esto lograría resolver el problema; empero como este proceso aumenta mucho el contraste entonces en algunas imágenes, sobre todo en aquellas con un fondo rugoso, la segmentación producía una saturación en el número de contornos o en el número de puntos que permite manejar el sistema BIOCOM, (imagen saturada visible en la figura 4.7). Este número es de 3600 contornos y de 10,000 puntos, respectivamente. La saturación se debe a que con el afinamiento de la imagen aparecen mayores objetos en una imagen con artefactos o ruido, es decir los artefactos es resaltada gracias al afinamiento.

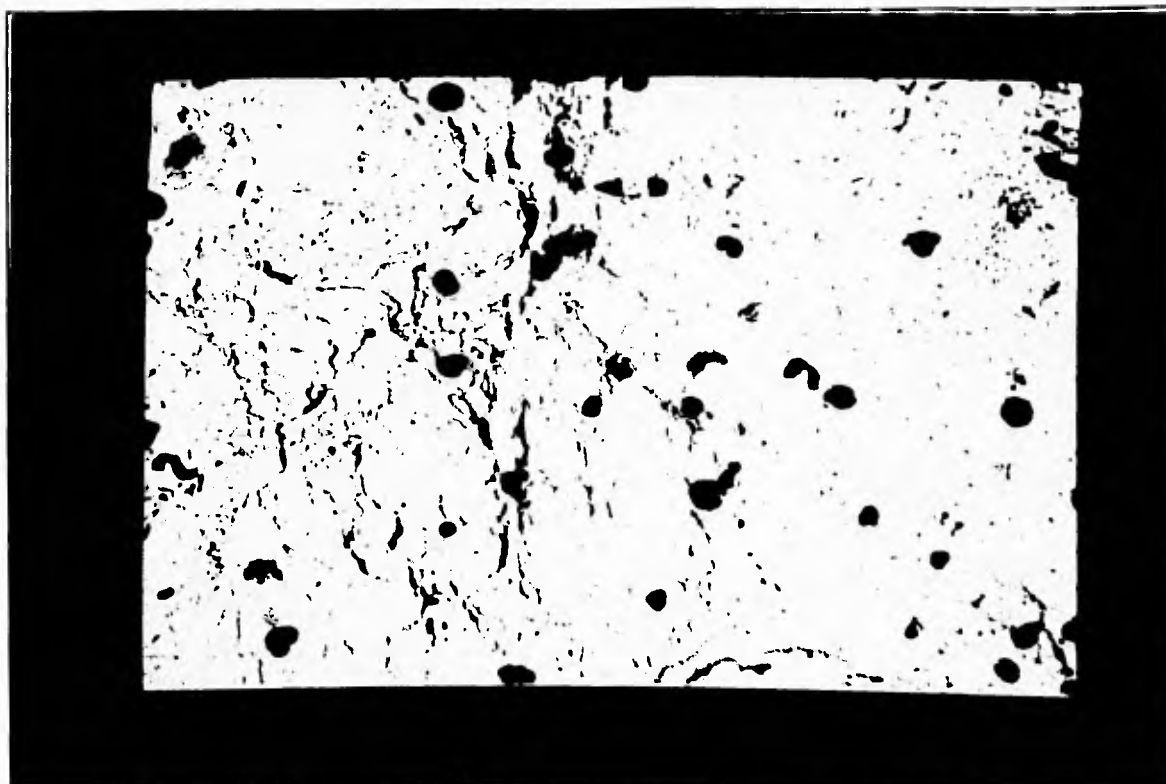


Fig. 4.7. Imagen saturada.

Se presentaban las siguientes alternativas: primera, el uso del afinamiento para resaltar algunos contornos y obtener de manera más precisa los núcleos en la pantalla pero la mala segmentación de los objetos en algunas pantallas. Segunda la inclusión de artefactos al no afinar la imagen pero una menor probabilidad de saturación en la segmentación. De ambas formas se presentaba una pérdida en la información.

La experimentación demostró que era factible realizar la segmentación de la imagen cuando aún no se le realiza el afinamiento (para posteriormente aplicarlo con el segundo método de afinamiento y sobreponer los contornos obtenidos anteriormente). Esto fue fácil de hacer pues el manejo de las tablas de contornos y el procesamiento de la imagen se manejan independientemente, afectando zonas de memoria distintas; además mientras no se renueven las tablas de contornos estos permanecen en memoria permitiendo afinar la imagen para después trabajar con los contornos sobre la imagen afinada. Este método permitió no modificar las medidas de forma y tamaño, lo cual era una condición establecida. Finalmente la forma de hallar los contornos fue la siguiente:

1. Segmentar sobre la imagen original P
2. Afinar la imagen original P y obtener la imagen P'
3. Sobreponer los contornos obtenidos en el paso 1 sobre la imagen original P

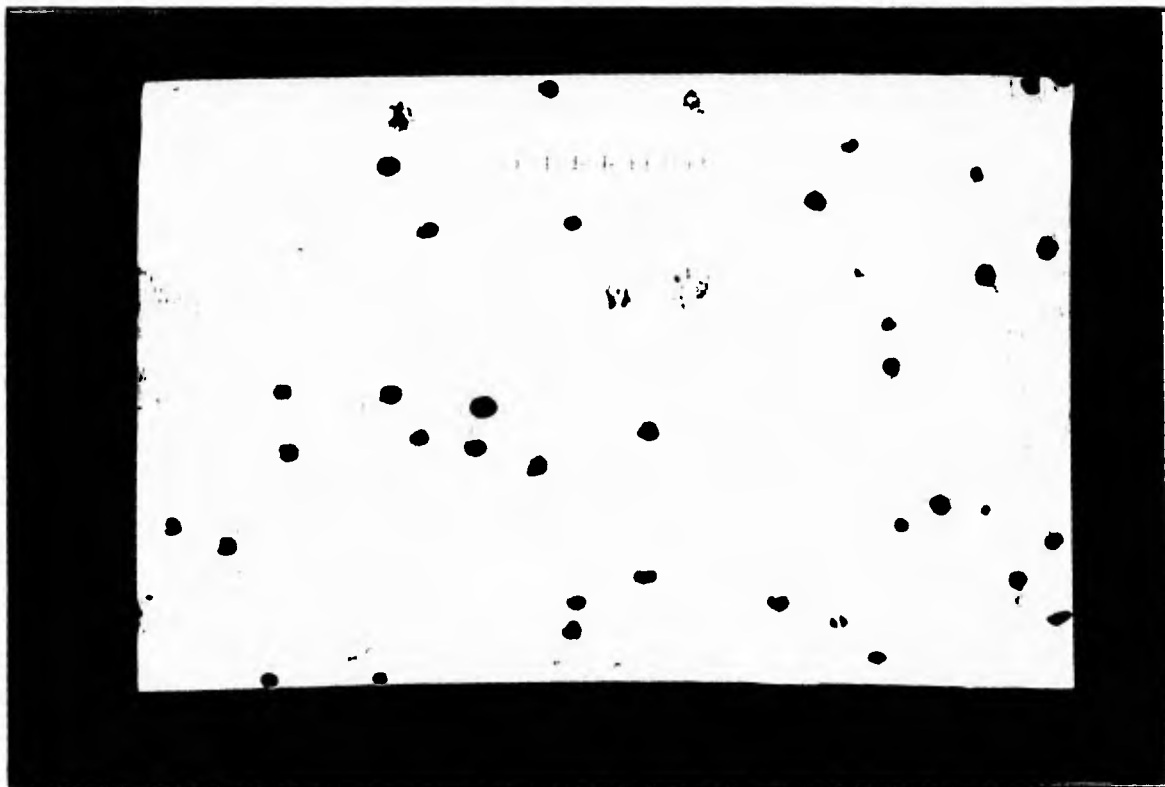
Con el método anterior se logró medir el nivel de gris y la desviación de niveles de gris en los contornos de forma tal que permitía un mejor aislamiento de los contornos que

representan núcleos estimulados. Sin embargo, posteriormente y debido a la experimentación, se retiró la medición del nivel de gris pues el tamaño de los núcleos está asociado a su color.

IV.3.2. Mitosis Compactas

Una vez resuelta la forma de reconocer los núcleos, el paso siguiente fue hallar el mejor método para encontrar las mitosis. Por su dificultad se decidió proseguir con las mitosis compactas pues no se tenía idea de como hacer su búsqueda.

La búsqueda de características comenzó por los parámetros más sencillos: factor de forma, nivel de gris, desviación estándar en el nivel de gris y área. Sin embargo, el factor de forma no es de mucha ayuda debido a que las mitosis compactas no se presentan con un patrón de forma fijo, incluso pueden variar de casi circulares a fusiformes (ver figura 4.8). Por principio se desechó este parámetro.

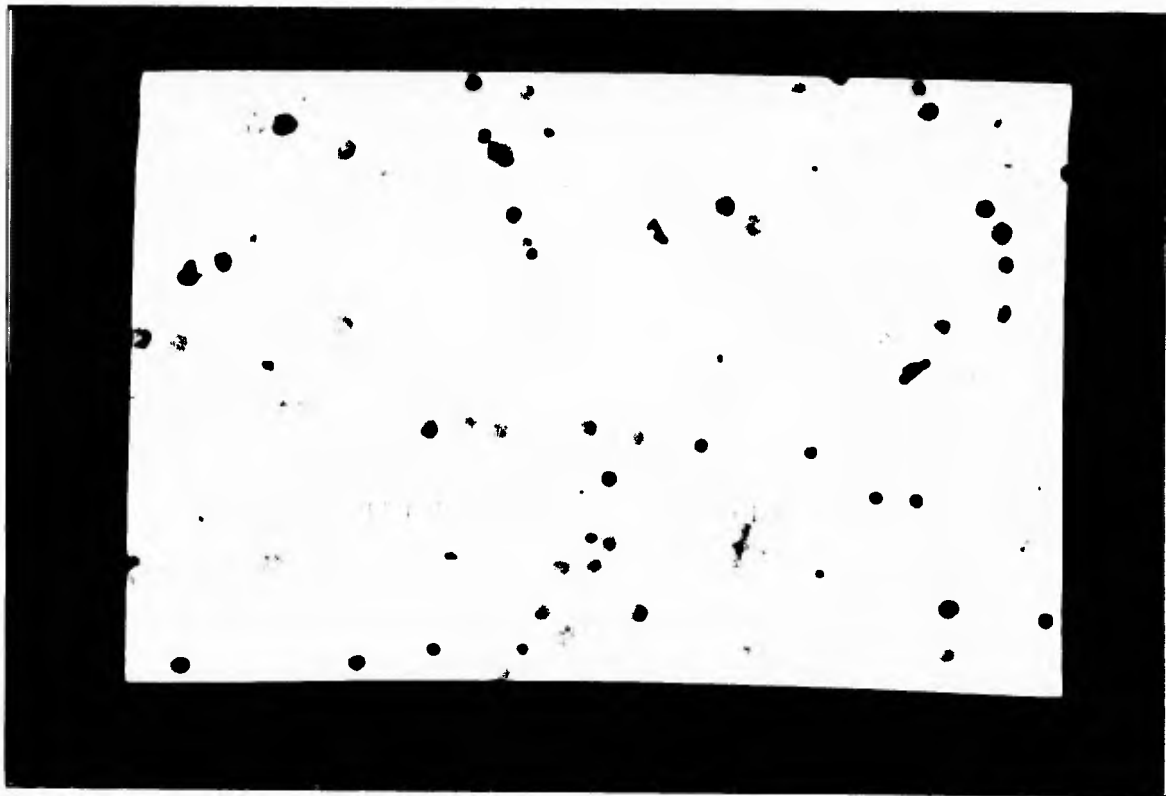


a)

Fig. 4.8. Mitosis compactas. a) ejemplo 1.
b) ejemplo 2. c) ejemplo 3.



b)



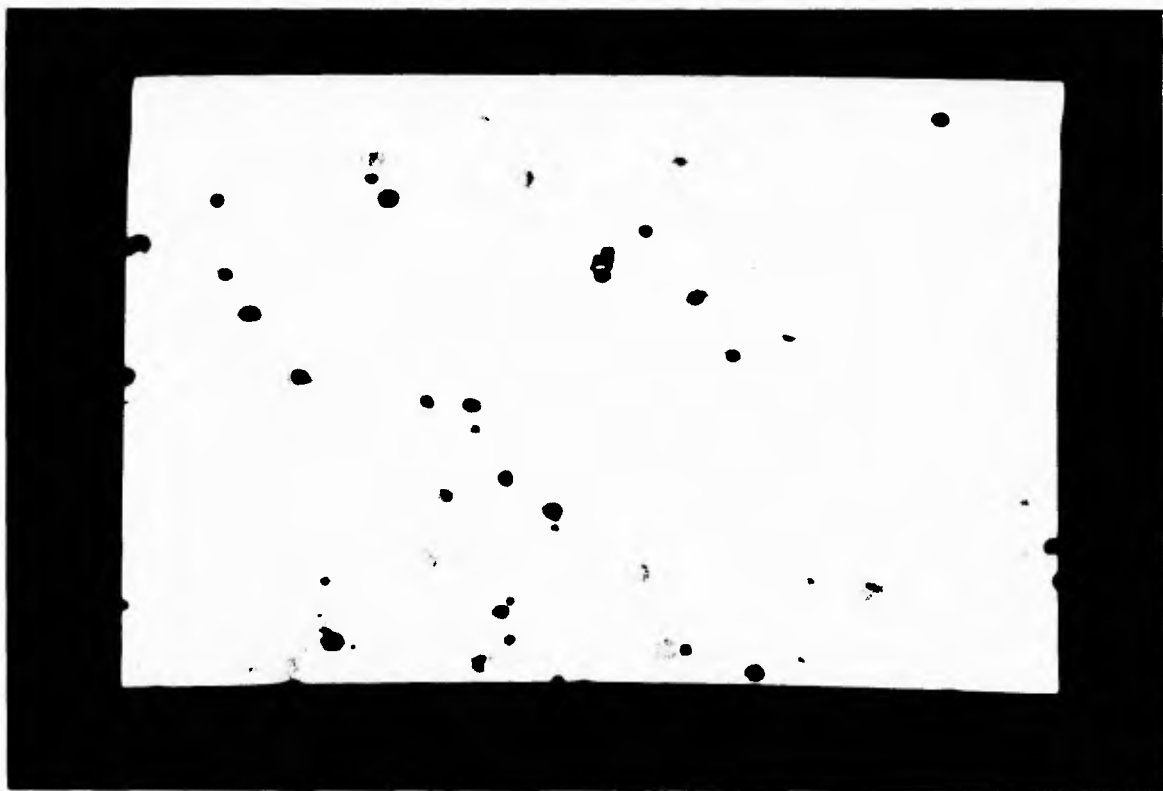
c)

Fig. 4.8. Mitosis compactas. a) ejemplo 1.
b) ejemplo 2. c) ejemplo 3.

Si se usa solamente el valor de nivel de gris, desviación estándar del nivel de gris y área las mitosis compactas pueden confundirse con núcleos estimulados o con artefactos. Es decir estos parámetros no son suficientes para lograr la extracción de contornos representando a las mitosis compactas de los demás. Es necesario hacer notar que el procedimiento para encontrar los contornos de las mitosis compactas fue el mismo que se realizó para los núcleos estimulados: primero segmentar sobre la imagen que proviene del microscopio y después afinarla para sobreponer los contornos obtenidos.

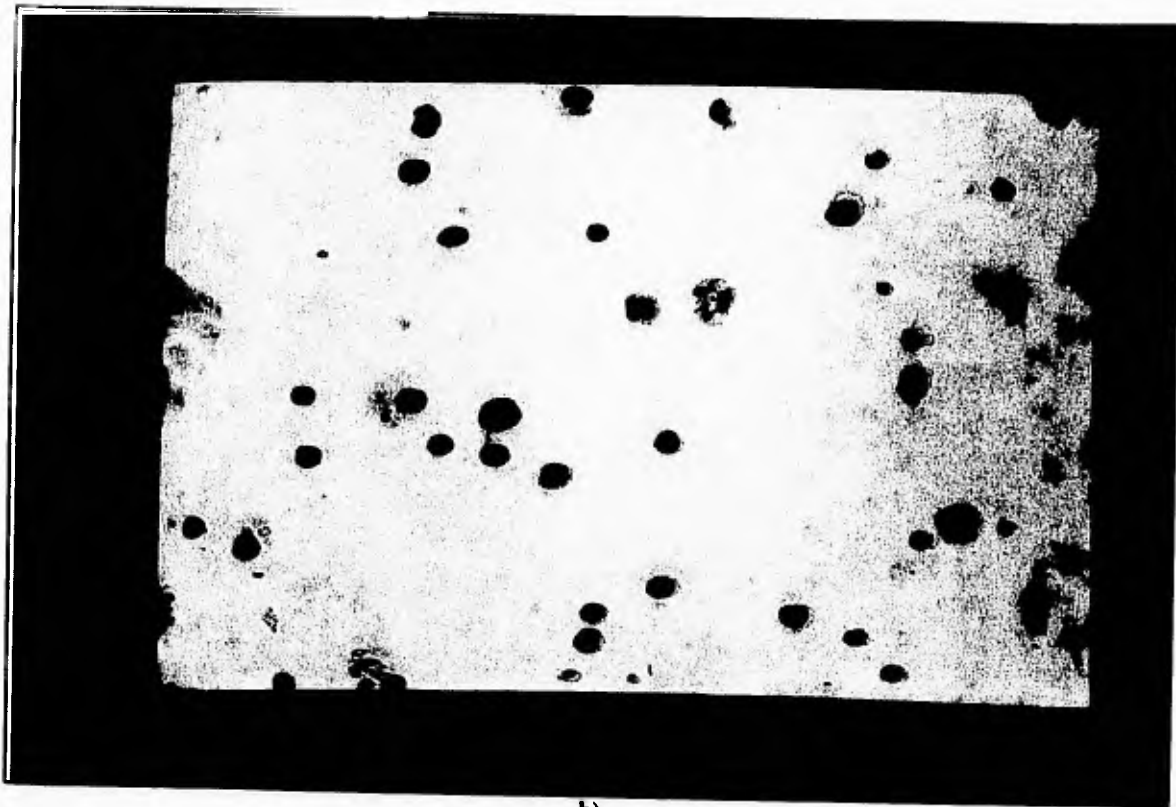
Las primeras búsquedas llevaron a ver que la segmentación de la imagen creaba contornos que encerraban niveles de gris, pertenecientes al fondo, dentro de ellos (ver figura 4.9). Esta situación modificaba enormemente las mediciones dentro del contorno, como el nivel de gris medio y su desviación estándar.

La búsqueda de una segmentación que no contuviera niveles de gris del fondo comenzó por hacer a los contornos más pequeños por medio de la modificación de las tablas que los definen.



a)

Fig. 4.9. Imagen con contornos conteniendo fondo en ellos. a) ejemplo 1.
b) ejemplo 2.



b)
Fig. 4.9. Imagen con contornos conteniendo fondo en ellos. a) ejemplo 1.
b) ejemplo 2.

Debido a que algunos contornos no tienen sus límites ajustados a los bordes de los objetos, las mediciones sobre los píxeles del contorno contienen ruido, es decir, se realiza la cuenta de elementos que no pertenecen al contorno. Para evitar esta pérdida de información se creó una rutina que permitiera reducir el contorno en ciertas unidades; es decir, erosionar el contorno. Por experiencia se dejó de lado la operación de afinamiento de la imagen antes de llevar a cabo la segmentación de la misma.

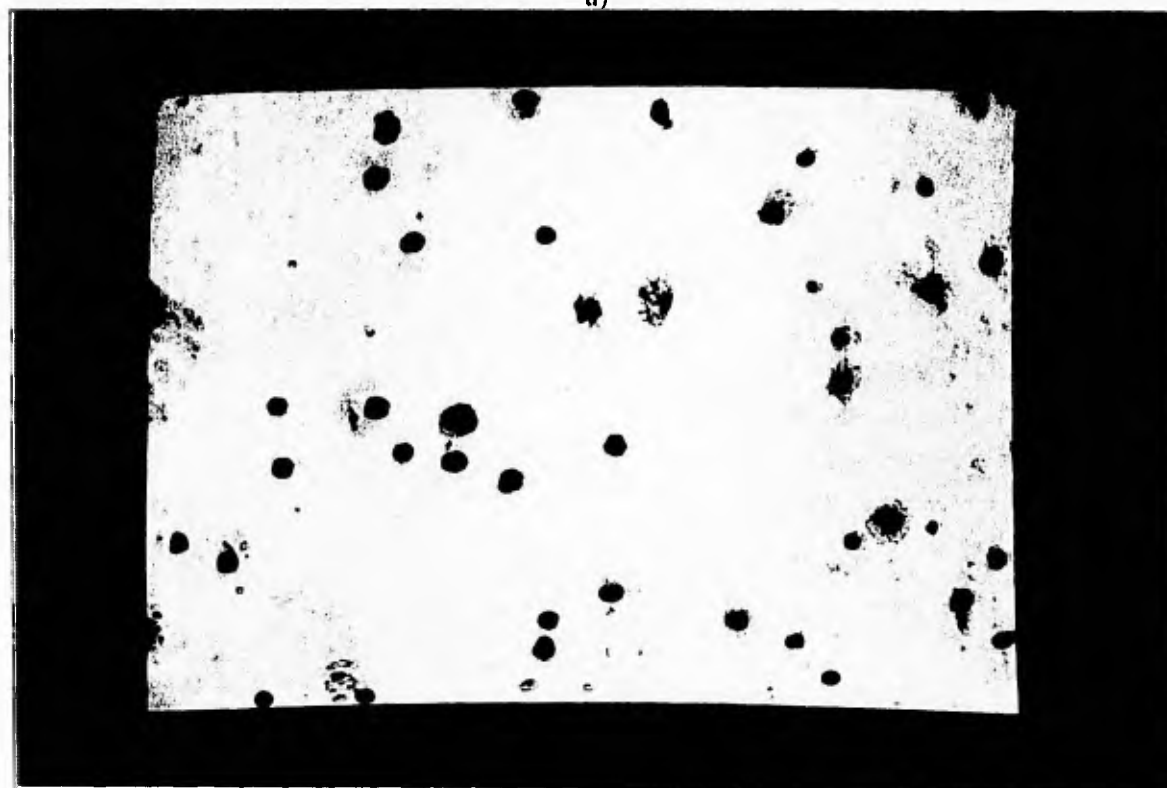
Para poder crear la función que redujera los contornos en un cierto número de unidades se tomó como base una función de BIOCROM que permite el acceso a las tablas de contornos. Además calcula el sentido de los ángulos existentes entre los puntos que definen el contorno y su centro de gravedad. La medida del ángulo es muy importante debido a que por medio de esta información se sabe si el contorno tiene una curva convexa o cóncava. Este tipo de curva permite saber si se suma o resta de las tablas de contornos el valor de unidades a erosionar (disminuir) sin que se modifique la forma del contorno. La subrutina que realiza esta operación se llama **EROSI**.

Sin embargo el uso de la función de erosión mostro que si se le quitan muchas unidades al contorno con respecto al centro de gravedad puede suceder que el contorno

cambie radicalmente su forma y se convierta en un contorno no convexo, como puede verse en la imagen siguiente.



a)



b)

Fig. 4.10. Proceso de reducción de un contorno. a) Contornos originales. b) Contorno colapsado.

Para evitar este problema se hizo un estudio para saber de cuanto era la media de reducción necesaria para que un contorno no fuera modificado en su forma. El estudio arrojó como resultado que era posible reducir 1.5 unidades siempre y cuando el contorno inicialmente tuviera una superficie mínima de 14 unidades. Prácticamente se erosiona 1.0 unidades cada vez que se usa esta función.

Esto solucionaba, en un alto porcentaje, el problema de los contornos incluyendo parte del fondo. Desde este momento, las medidas de nivel de gris y desviación estándar del nivel de gris dentro del contorno serán tomadas después de la erosión. Empero seguía sin producirse ningún resultado que encontrara las mitosis compactas y que no incluyera núcleos o artefactos usando los mismos parámetros de los núcleos estimulados. La búsqueda de parámetros debía continuar.

IV.3.2.1. Textura

La forma en que el ojo humano reconoce cuerpos como las mitosis compactas es por medio de la textura, entonces era preciso hallar parámetros de textura dentro de los contornos que pudieran ser mitosis compactas.

Sobre medidas de textura existe mucho escrito, aunque en realidad estos métodos requieren de un gran número de operaciones o de otros recursos, como memoria de la máquina. Es indispensable tener siempre en cuenta que en un sistema de este tipo el tiempo de proceso es importante, pues si el sistema es excesivamente lento, por lo que es preferible seguir haciendo los cálculos manualmente.

Partiendo del hecho anterior, era indispensable hallar medidas de textura que fueran fáciles de calcular pero que ayudaran a identificar a los contornos que pudieran contener mitosis compactas. La búsqueda se prosiguió siguiendo el paralelismo que las imágenes tienen con los relieves. Los niveles de gris significan de alguna forma profundidades y picos.

Si fuera posible tener un nivel medio, algo como el nivel del mar, que sirviera como referencia para medir el número de picos, valles y depresiones; entonces, se podría tener un parámetro de textura bastante rápido. Esto es posible debido a la forma en que trabaja la rutina que permite saber que pixeles pertenecen a un contorno.

Por medio de la rutina `fg_rowr`, de MATROX, es posible leer los pixeles que pertenecen a una línea y por medio de otra rutina de BIOCOTM definir los que pixeles están dentro del contorno. Las rutinas de lectura y escritura de pixeles de MATROX son muy rápidas en su funcionamiento.

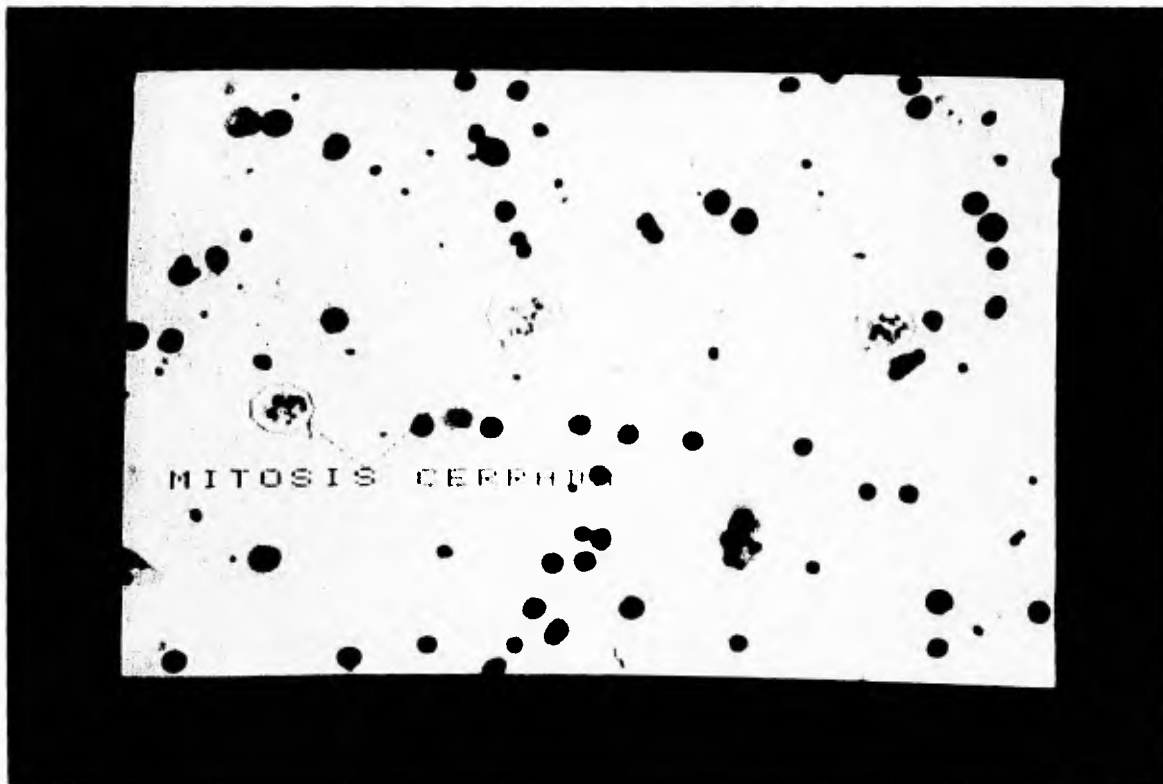
Al parecer, la mejor forma de hallar una referencia dentro de un contorno era un nivel medio de clases internas; es decir, un proceso Otsu al interior de cada contorno.

Este fue el siguiente paso: realizar una función que obtuviera el valor de división de clases dentro de cada contorno. La función que realiza esta operación es **OTSU**, en FORTRAN, que usa como base la función **ROUTI**, la cual hace referencia a los pixeles que pertenecen exclusivamente a un contorno y a **find_umbral**.

En este momento se tenía la referencia y el acceso a cada pixel dentro del contorno, faltaba buscar la mejor forma para detectar las depresiones, valles y picos.

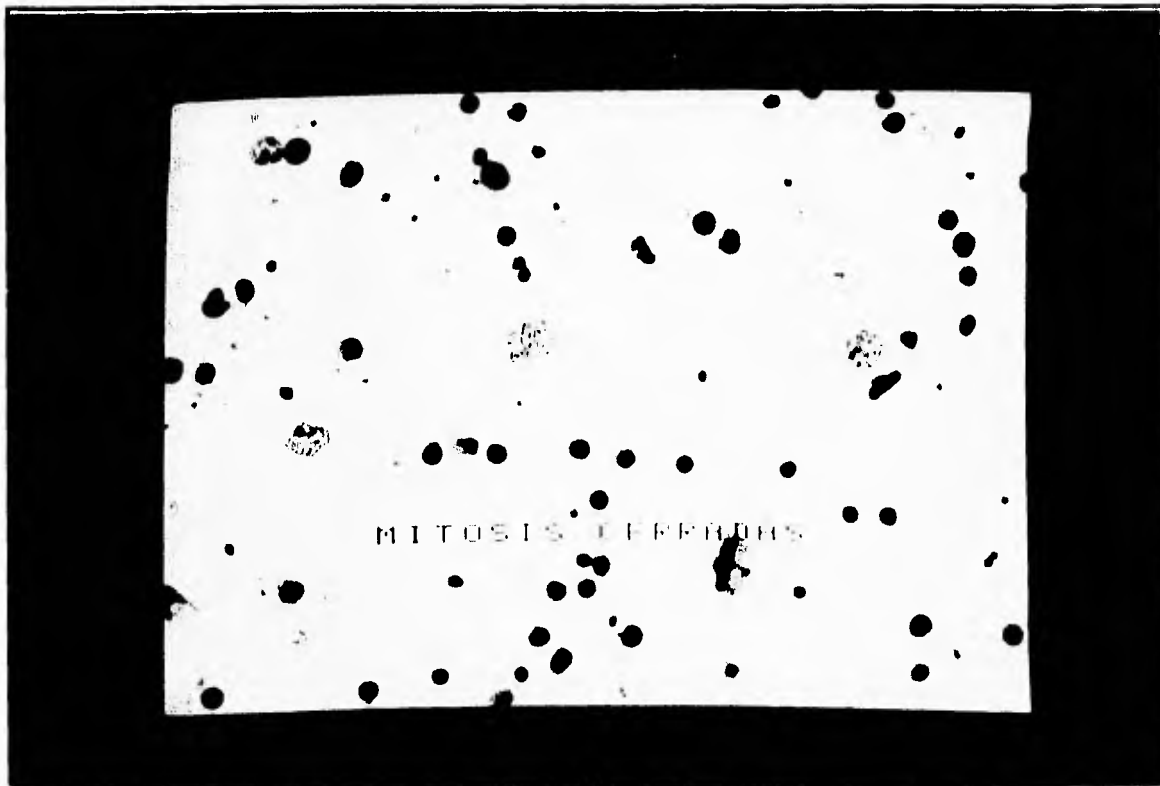
Desde el principio se decidió medir el número de cambios que se presentaban con referencia al nivel patrón (Otsu) en el total de pixeles que conformaban el contorno.

Otra vez este parámetro se quedó corto con la exigencia como parámetro definitivo para la identificación de las mitosis compactas, debido a que al hacer la relación de los cambios de nivel de gris entre el número total de pixeles la medición equivalía a la media de cambios. Como se sabe la media no siempre da la mejor información sobre lo que está sucediendo en una población. Por ello era necesario medir los cambios de otra manera que no tomará mucho tiempo.



a)

Fig. 4.11. Mitosis compactas segmentadas. a) ejemplo 1.
b) ejemplo 2.



b)

Fig. 4.11. Mitosis compactas segmentadas. a) ejemplo 1.

b) ejemplo 2.

Se decidió hacer una nueva investigación sobre cómo se realizaban las mediciones dentro de la rutina **ROUTI**. Estas son hechas hilera por hilera e identificando de cada fila horizontal aquellos pixeles que pertenecen al contorno (ver rutina **CONTRA** y figura 4.12). Entonces fue cuando surgió la idea de medir el número de cambios que suceden por cada línea.

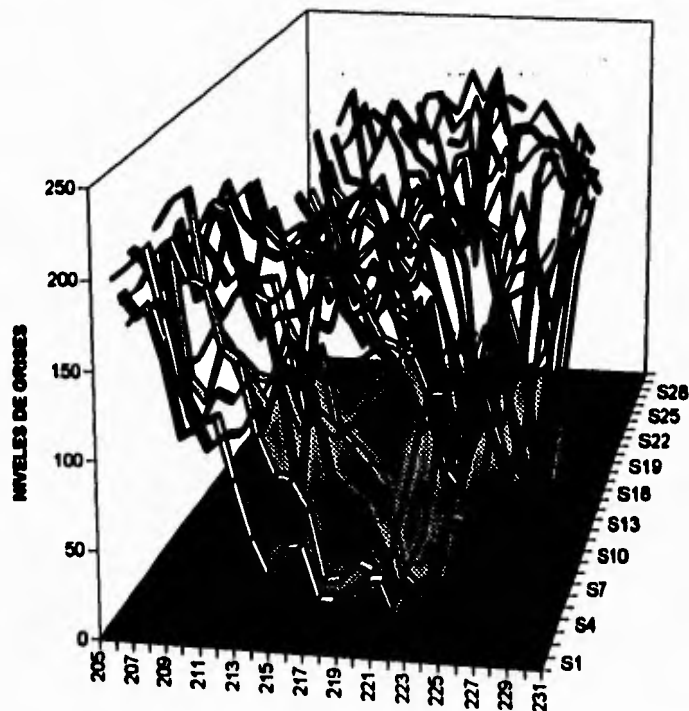


Fig. 4.12 Esquema de la forma en que se miden las líneas dentro de un contorno.

Aunque pareciera que el conteo de cambios por línea debiera ser igual al de la medición de cambios por píxeles, en realidad hay una diferencia debida a que los cambios por píxeles es una medida de la media de cambios dentro del contorno mientras los cambios por línea da una noción del comportamiento en cada hilera del contorno. Esta diferencia es significativa en los casos en donde hay huecos claros dentro del contorno porque aumenta el número de cambios por hilera. Aunque la medida de cambios en el total de píxeles parecía obsoleta, en realidad es un parámetro que permite tener idea de la densidad de cambios en las mitosis compactas, así que se mantuvo como parámetro.

Otra medición que se decidió tomar fue la del nivel de contraste dentro de un contorno, que también es una medida de la textura de un objeto. La forma en que se midió el contraste interno fue por medio de la misma referencia, nivel de gris Otsu, y calculando el promedio de niveles de gris por encima y por debajo de dicha referencia relacionándolos por medio de una resta. Todas estas mediciones pueden obtenerse por medio de la subrutina **CONTRA** que, nuevamente, toma como base la rutina **ROUTI** para hacer referencia a los píxeles que pertenecen a un contorno.

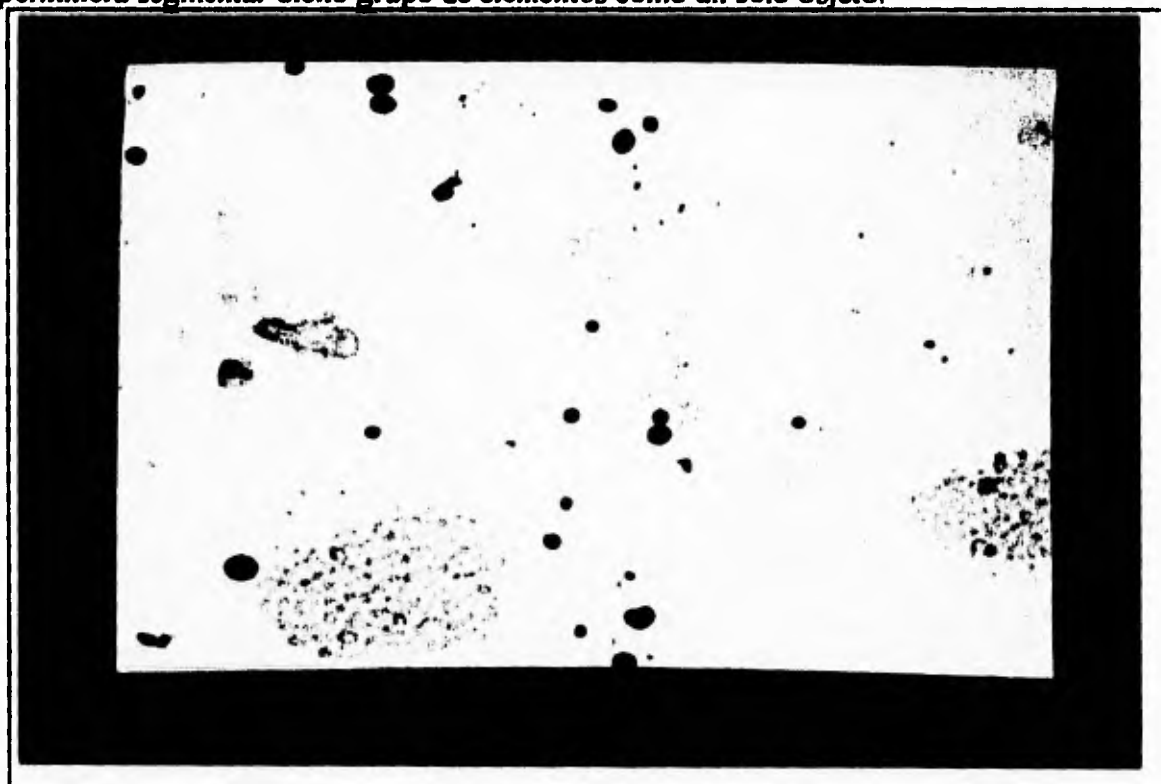
Las mediciones de textura pueden ser limitadas si únicamente se hacen en sentido horizontal, así que se llevó a cabo la implementación que permitiera la medición de cambios

por línea y por pixel en sentido vertical. Estas mediciones también se realizan por medio de la rutina **CONTRA**, aprovechando las mediciones que se hacen para medir de forma horizontal.

IV.3.3. Mitosis Dispersas

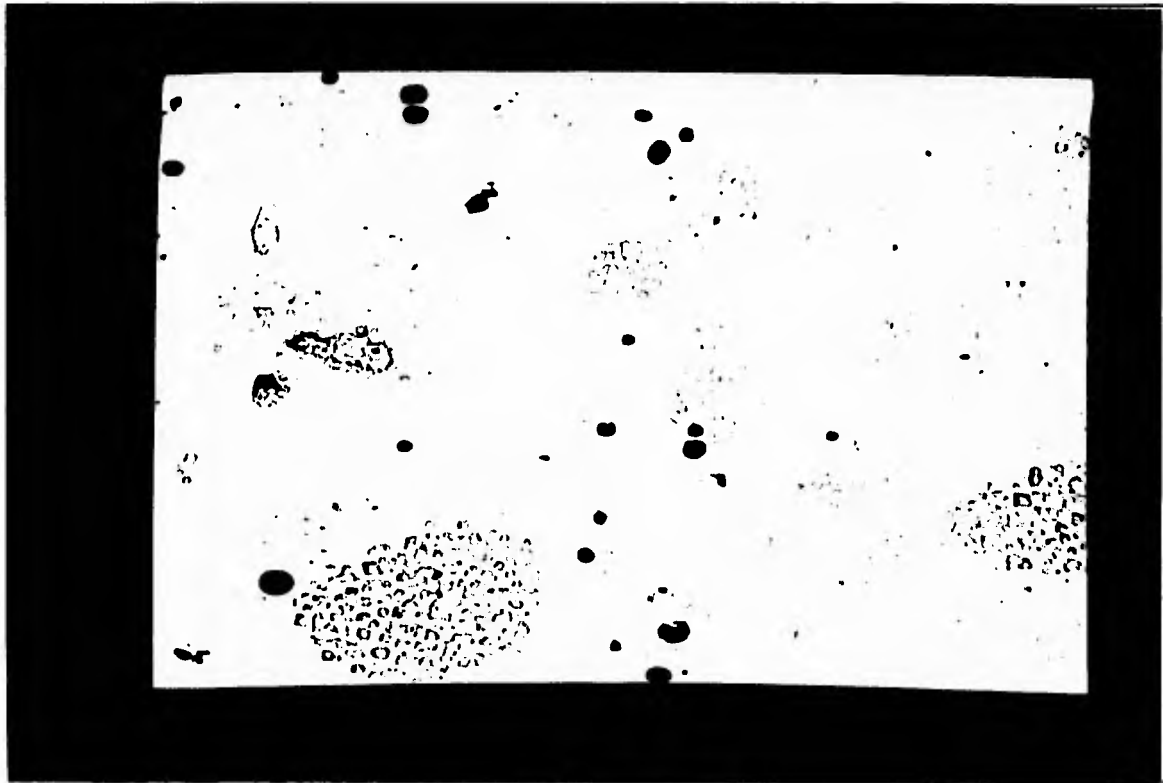
Por último quedaban las mitosis dispersas. La primera dificultad para hallar las mitosis dispersas se encuentra en poder unir los cromosomas esparcidos en un solo cuerpo. Los intentos para lograr el agrupamiento de cuerpos se dirigieron en dos sentidos; el primero, fue hacer un cálculo de la cercanía de los centros de gravedad de los contornos, y el segundo era hallar un procesamiento no morfométrico que facilitara la obtención del grupo de cromosomas antes de la segmentación.

El primer método consistió en calcular las distancias entre cuerpos y, por medio de una referencia definida por el usuario mediante un entrenamiento (la cuestión del entrenamiento se tratará más adelante), decidir si un contorno pertenecía a un agrupamiento o no. Después de identificar aquellos elementos del grupo sería necesario, para obtener sus características morfométricas, modificar las tablas de contornos para obtener un solo contorno que incluyera a todos los elementos, o bien realizar un proceso a la imagen que permitiera segmentar dicho grupo de elementos como un solo objeto.

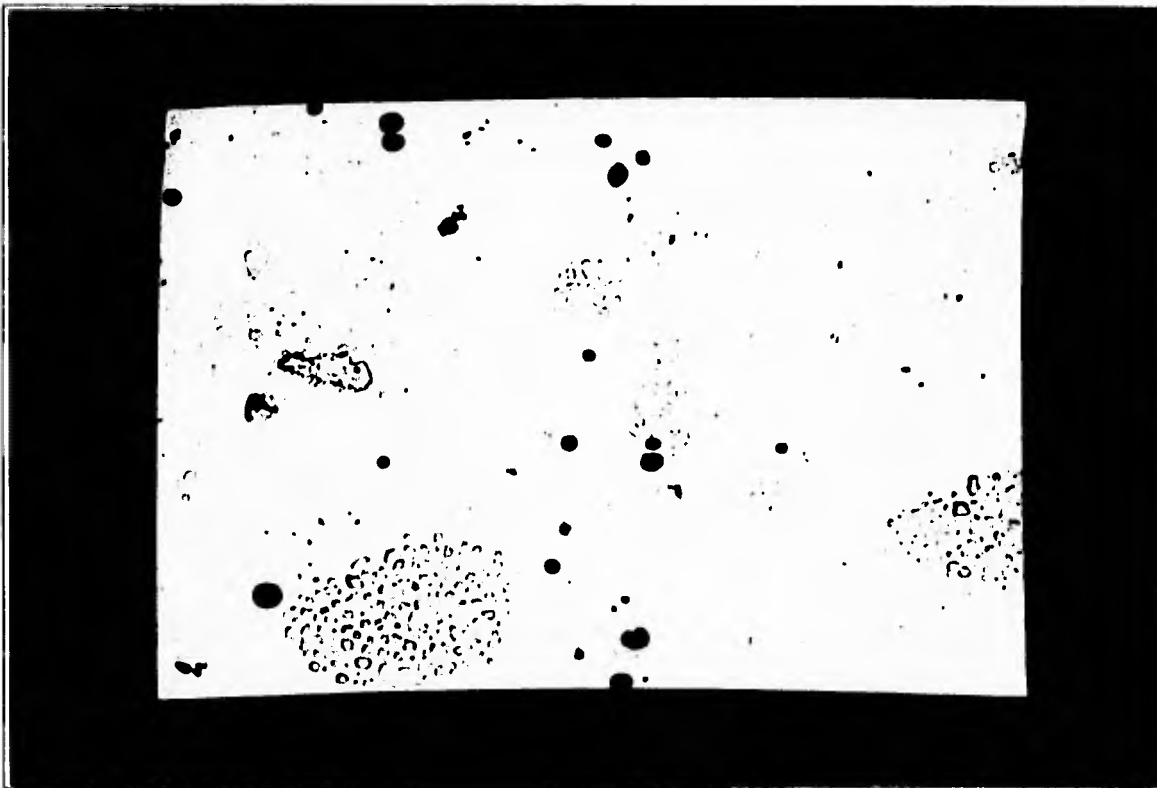


a)

Fig. 4.13. Proceso de aglomeración de objetos. a) Imagen inicial. b) Segundo paso.
c) Resultado del método.



b)



c)

Fig. 4.13. Proceso de aglomeración de objetos. a) Imagen inicial. b) Segundo paso. c) Resultado del método.

Realmente la identificación de contornos cercanos no significaba gran problema ya que BIOCUM permite obtener el centro de gravedad de cada contorno por medio de la

función ROUTI y el cálculo de distancia es muy simple: $\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$

donde (x_1, y_1) son las coordenadas de uno de los centros de gravedad de un contorno y (x_0, y_0) las coordenadas para otro centro de gravedad, así que la identificación de los elementos del grupo no acarrearón ningún problema; sin embargo la modificación de las tablas de contornos para que todos los elementos hallados conformaran un cuerpo representaba bastante tiempo y esfuerzo, así como de una gran complicación en el algoritmo de modificación de las tablas que definen los contornos.

Ante la necesidad de hallar un proceso tal que modificara la imagen para que aparecieran los elementos como un solo objeto, ¿por qué no hacerlo desde el principio y ahorrarse el cálculo de distancias?. Era cierto que hacer el cálculo de distancias provocaba un mayor gasto de tiempo, el cual podía ser ahorrado si se procesaba la imagen desde el inicio.

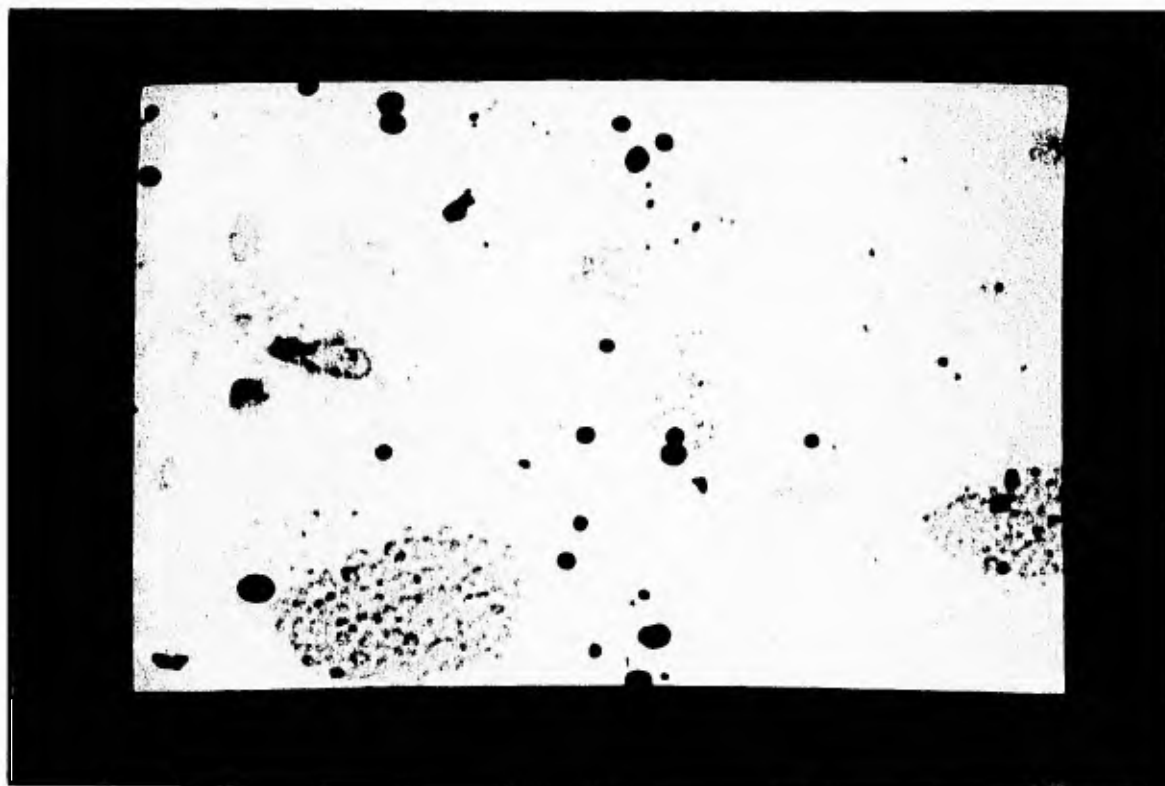
De esta manera se abandonó la idea de calcular las distancias y se decidió continuar por la segunda vertiente: hallar un proceso que permitiera conjuntar a los cromosomas de una mitosis dispersa antes de segmentar la imagen, y obtener un solo contorno de ellos.

La idea era más o menos clara, había que hacer crecer a los objetos de la pantalla de tal forma que se unieran con sus vecinos y formaran un solo cuerpo. Esto era bastante sencillo, pues dilatando la imagen sucedía este efecto en los objetos dentro de la pantalla. Sin embargo, la primera opción fue dilatar y después erosionar la imagen, proceso que se conoce como cerradura. No se debe confundir el proceso de erosión de la imagen con el de erosión de los contornos, son diferentes.

La cerradura permite desaparecer los huecos, defectos y divisiones internas en un grupo de píxeles con ciertas características, o sea a los objetos en la pantalla.

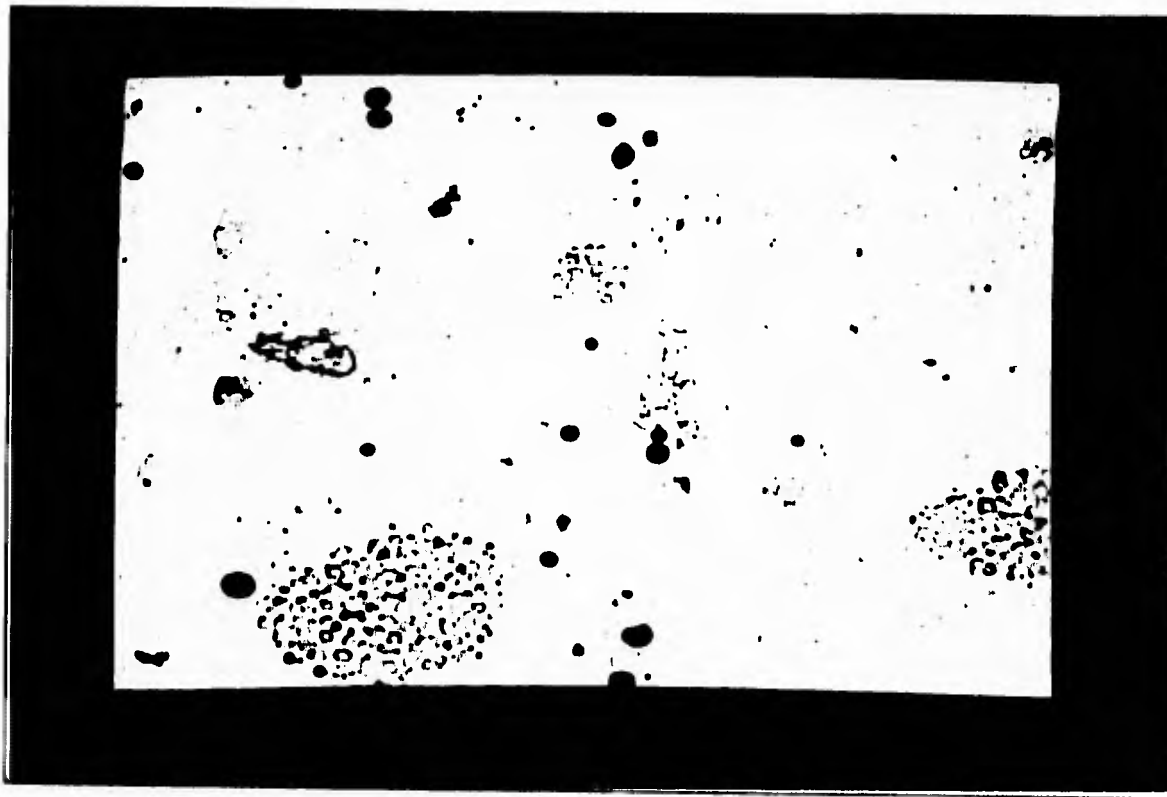
El primer problema era saber cuántas dilataciones y erosiones eran necesarias para abarcar a todos los cromosomas en un solo cuerpo; pues como ya se mencionó es posible hallar cromosomas poco esparcidos, así como aquellos que llenan buena parte de la pantalla (ver figura 4.2). Por lo tanto se llevó a cabo otro estudio para averiguar el número de dilataciones necesarias para abarcar la mayoría de las mitosis dispersas. El estudio condujo a saber que el número promedio, efectivo, de dilataciones era de dos. Esta operación puede ser vista en la figura 4.13.

El uso de la operación de cerradura tiene el inconveniente, que al hacer crecer y después reducir a los objetos, se pierden las características de textura de las mitosis dispersas, como se muestra en las siguientes imágenes.

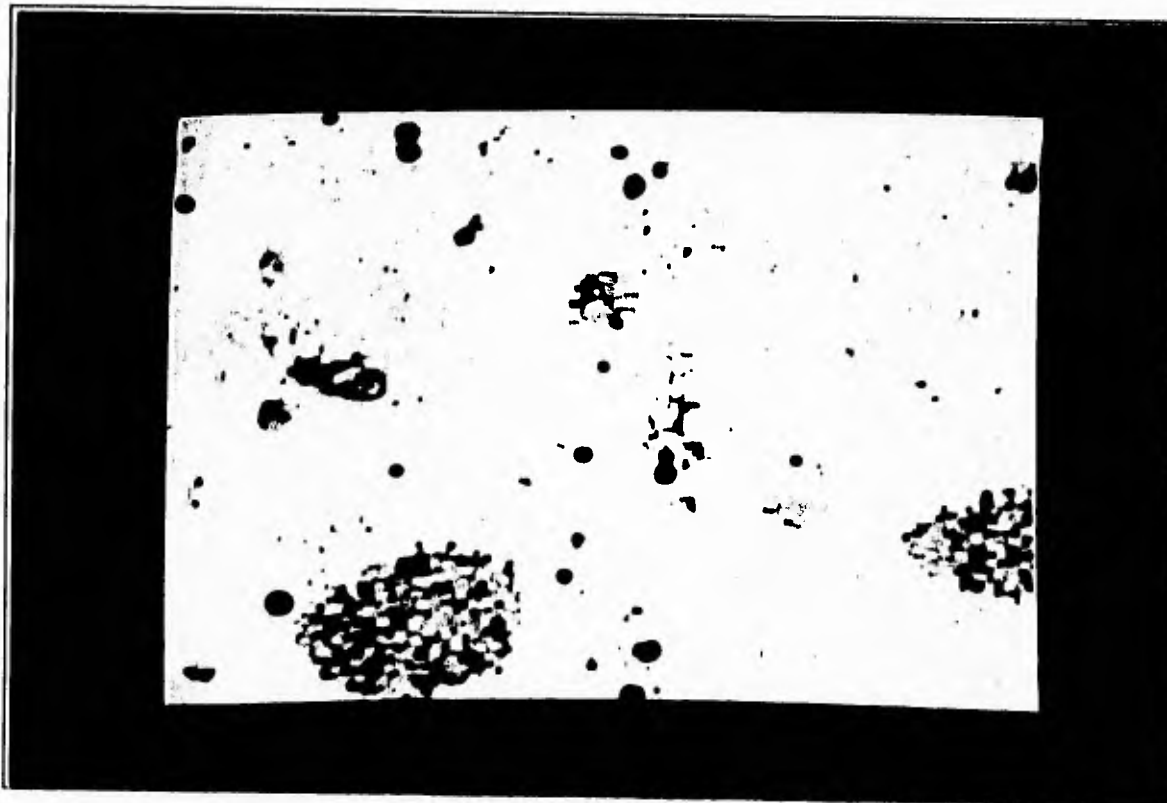


a)

Fig. 4.14. Pérdida de características en operaciones de apertura. a) Imagen original
b) ejemplo 2. c) ejemplo 3.



b)



c)

Fig. 4.14. Pérdida de características en operaciones de apertura. a) Imagen original
b) ejemplo 2. c) ejemplo 3.

Está pérdida de características a veces era acompañada de pérdida de los límites de las mitosis dispersas, así que se decidió buscar otra alternativa.

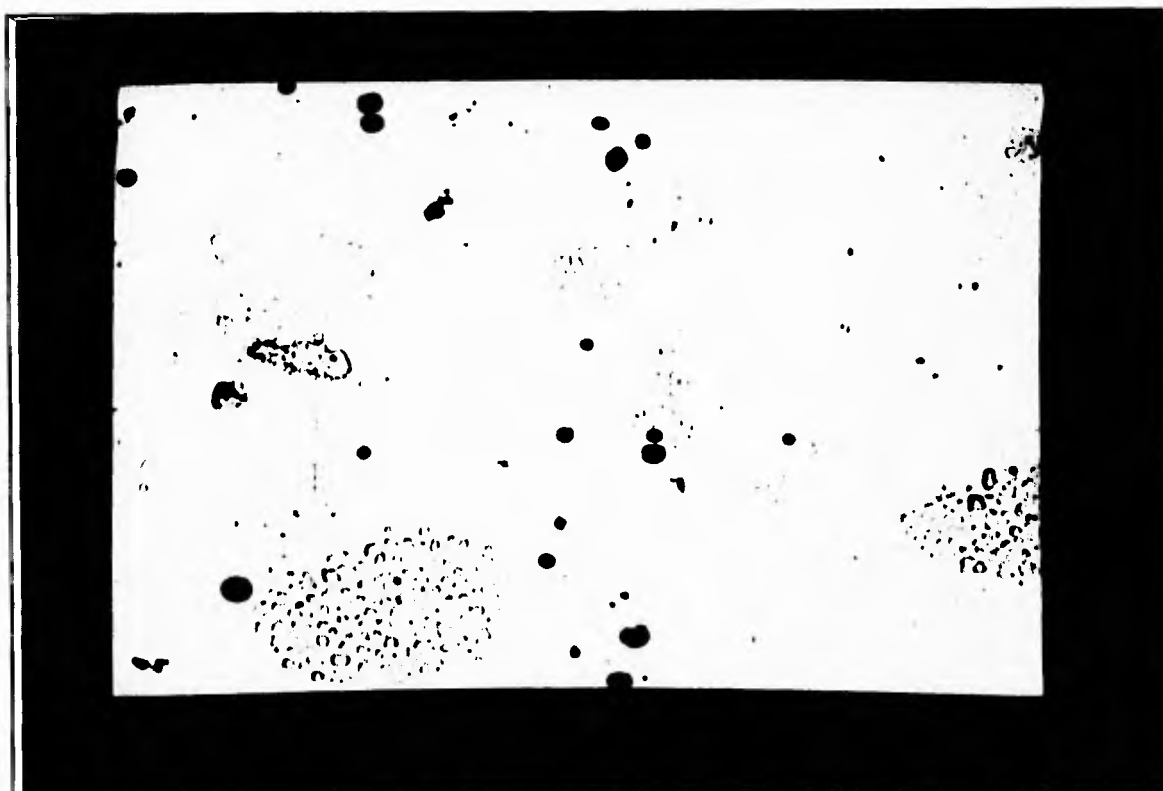
La siguiente alternativa fue la de aplicar únicamente dilataciones pues ya se tenía el estudio del número de dilataciones necesarias para abarcar la mayoría de los cromosomas que forman parte de las mitosis dispersas, así que únicamente quedaba hacer pruebas con esta hipótesis.

Se probó segmentar sobre la imagen doblemente dilatada lo que produjo contornos abarcando al total de los cromosomas de cada mitosis dispersa. Después de realizar la segmentación se intentó obtener características de las mitosis dispersas en esta imagen; sin embargo dichos objetos en la pantalla parecían perder características que los resaltarán de los demás objetos, tanto artefactos como mitosis compactas y núcleos. Al crecer los píxeles invaden otros píxeles perdiéndose cierta información de la imagen. Ante esta pérdida se decidió tratar de sobreponer los contornos obtenidos en la imagen doblemente dilatada en la imagen original.

Como se mencionó, las tablas de contornos son trabajadas de manera separada a las imágenes, así que es posible segmentar en una imagen y luego cambiar la imagen para posteriormente poner los contornos obtenidos de la primera imagen en la segunda. Este fue el proceso que se siguió, primero se dilató la imagen dos veces con la rutina `fg_erode` y sobre esa imagen se realizó el proceso de segmentación. Este proceso puede verse en la secuencia siguiente.



a)



b)

Fig. 4.15. Proceso de segmentación de mitosis dispersas. a) Imagen original. b) segundo paso.
c) tercer paso. d) Resultado.

Otra vez se contaba con los contornos, pero había que obtener las características de aquéllos que realmente fueran mitosis dispersas.

Nuevamente la forma en que las mitosis dispersas son reconocidas por el ojo humano es por medio del uso de textura, así que se usaron los parámetros que caracterizaban a las mitosis compactas.

El cálculo de área ayudaba bastante a discriminar los contornos que encerraban a las mitosis dispersas, pues estas poseen una superficie mayor que los demás cuerpos de interés, pero se presentaban artefactos que entraban como mitosis dispersas a pesar de los otros parámetros usados para clasificar a las mitosis compactas. Los parámetros que caracterizaron a las mitosis compactas no eran suficientes para solamente elegir las mitosis dispersas, así que había que buscar un nuevo parámetro.

La búsqueda derivó en muchas vertientes: procesos de afinamiento, uso de diferentes filtros en el microscopio, uso de histogramas de cada contorno, etcetera pero la búsqueda era infructuosa. Sin embargo, al hacer cálculos sobre el histograma de frecuencias acumuladas de los niveles de gris que están dentro de un contorno, se halló una forma característica de dicho histograma para las mitosis dispersas. Esta forma se presentaba en las mitosis dispersas y, prácticamente no se presentaba en ningún otro cuerpo. Esta forma del histograma puede ser vista en la siguiente imagen.

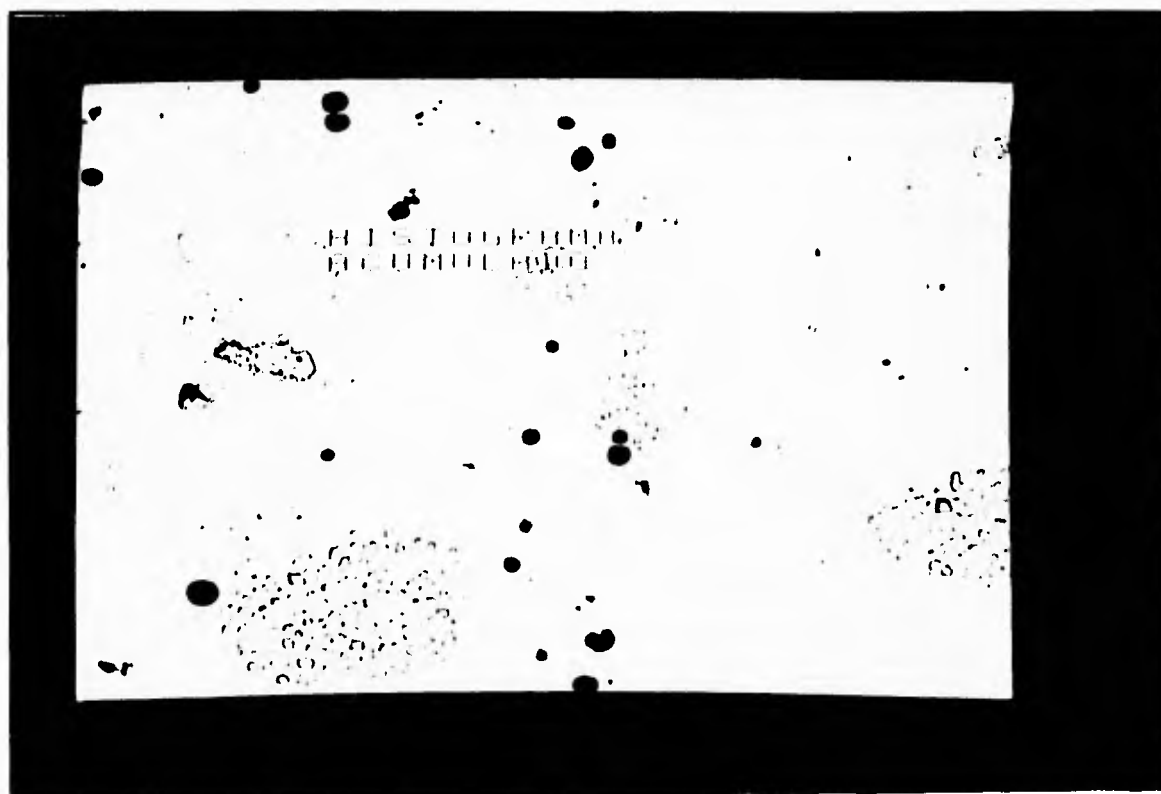


Fig. 4.16. Imagen del histograma acumulado característico de las mitosis dispersas.

Como puede verse en el histograma acumulado se presenta un cambio brusco de la curva. Este cambio se debe a que los contornos que se obtuvieron por el método de dilatación contienen información del fondo en abundancia y niveles de gris de los cromosomas en menor cantidad. El siguiente histograma muestra esta distribución en la forma antes mencionada.



Fig. 4.17. Zonas de grises de las mitosis dispersas.

El siguiente paso era poder lograr un método para que la máquina pudiera obtener esa misma información. El problema parecía igual de complicado que reconocer las mismas mitosis. Era necesario hallar un método que ayudara a reconocer la forma de una curva.

Esta idea desapareció al ver que se pueden notar dos zonas casi rectas dentro del histograma de frecuencias acumuladas, como se muestra en la figura 4.18.

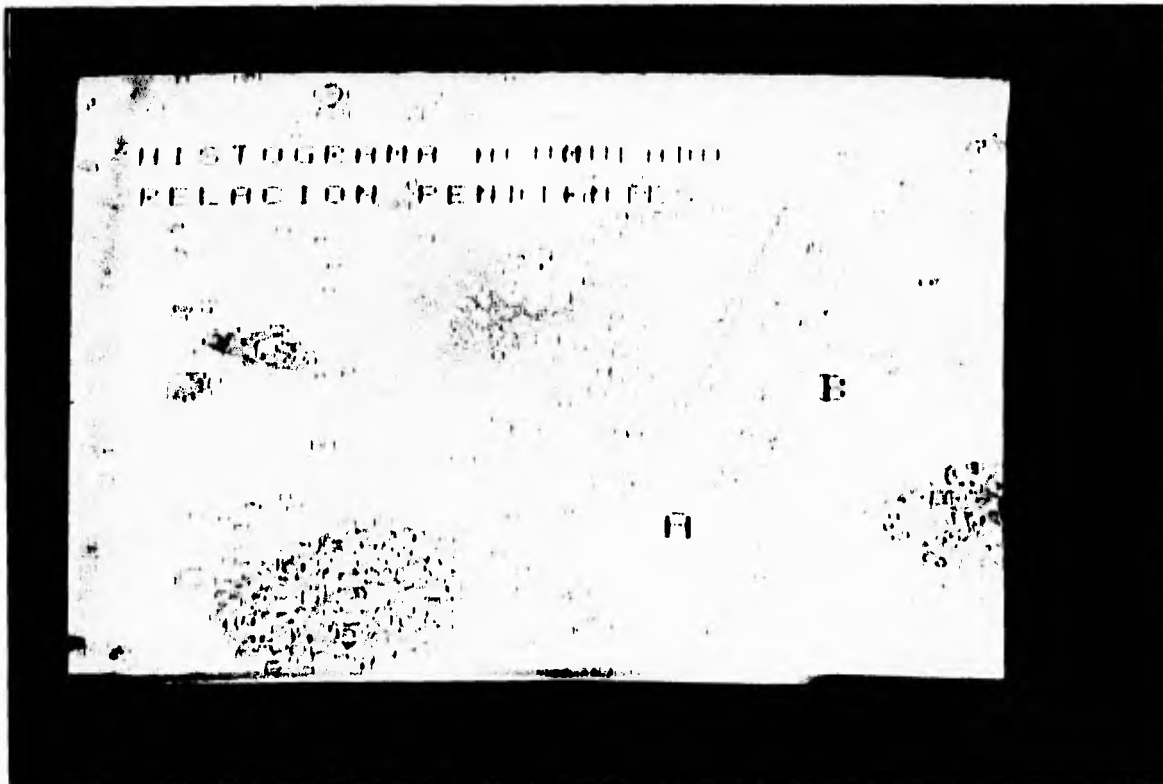


Fig. 4.18. Zonas rectas del histograma acumulado.

Esas zonas poseen pendientes que pueden ser relacionadas entre sí, dando como resultado un parámetro de como se está comportando el histograma de cada contorno candidato a mitosis dispersa.

Esa información estaba bien identificada pero era necesario encontrar los puntos en donde comenzaba y finalizaba cada sección recta.

En la imagen de cada histograma el fondo de la imagen representa el nivel de gris y la altura de la imagen representa la frecuencia de los niveles de gris dentro de la pantalla.

De nueva cuenta se vio que el punto de inicio de la primera zona recta es aquel que divide a los dos niveles de gris contenidos en el contorno. Otra vez el método de Otsu nos ayuda a descubrir dicho punto de inicio (P_0) así que se volvió a usar la subrutina **OTSU**, para este nuevo propósito.

Con el nivel de gris obtenido por OTSU se tiene el punto de inicio de la primera zona recta pero falta el punto de finalización (P_1) de la zona recta número uno (R_1) y el punto final (P_2) de la zona recta número dos (R_2). Experimentalmente se definieron los dos puntos que limitan a R_1 y R_2 usando como base el nivel de referencia obtenido con **OTSU**. Los límites quedaron como sigue: $P_1 = 1.25 * OTSU$ y $P_2 = 1.5 * OTSU$. Estos límites ofrecen una ventaja pues los tres puntos son dinámicos con respecto a la distribución de niveles de gris contenidos en cada contorno.

La subrutina que realiza este cálculo es **HISZON**, la cual está basada en la rutina de **BIOCOM** que calcula el histograma de frecuencias acumuladas de una zona de la pantalla. Como puede verse en esta implementación, la rutina requiere de que los puntos de referencia le sean pasados así que pueden ser pasados otros valores diferentes a los valores P_1 y P_2 antes definidos con respecto a **OTSU**, o también es posible cambiar el valor del punto P_0 por otro calculado de manera diferente. Sin embargo, la rutina siempre calculará la relación de la manera siguiente:

$$\text{Pendientes} = \text{ángulo}(P_0P_1) - \text{ángulo}(P_1P_2)$$

Se puede notar en la implementación que existe otro dato que resulta de **HISZON**. Este nuevo dato es un parámetro de textura derivado del mismo estudio del histograma de frecuencias acumuladas y se trata de la distribución de niveles de grises a partir del nivel medio de la pantalla. De alguna forma este parámetro nos da una idea del cómo se encuentra la densidad de niveles de gris en un contorno.

No obstante que este último parámetro es utilizado para hacer más fino el tamiz del filtro, el verdadero parámetro que ha ayudado a discriminar a los contornos que representan a las mitosis dispersas de los que no son es el de la relación de pendientes (se le llama así porque, aunque se calculan ángulos, las pendientes son una representación de los ángulos) haciendo bastante eficiente al método. El método de la relación de pendientes y la distribución de niveles de gris, aunados con los parámetros de textura usados para hallar mitosis compactas y los parámetros de área y desviación estándar del nivel de gris, permiten un reconocimiento con alto grado de certeza de las mitosis dispersas, casi 90%.

Como se vió que la distribución de niveles de gris otorgaba cierta información sobre la textura de los objetos, se decidió probar este parámetro en la búsqueda de mitosis compactas. Finalmente se observó que también restringía la entrada de artefactos y que no requería mucho tiempo para el cálculo así que se dejó definitivamente en la indagación de mitosis compactas.

Para finalizar con la forma en que se reconocieron las mitosis dispersas hay que decir que después de implementar todas las rutinas de reconocimiento se tuvo un último problema. Este problema fue que en algunas imágenes los cromosomas tenían un nivel de gris tan leve que aun al ser dilatados, seguían sin ser identificados como un contorno y se perdían con el fondo. Para evitar esto se volvió a hacer una investigación sobre cómo lograr segmentar dichos objetos.

Nuevamente había que aumentar el contraste de la imagen para que los cromosomas fueran identificados como un solo grupo, así que se intentó hacer un afinamiento de la imagen original antes de dilatarla. Como al dilatar la imagen disminuye el número de cuerpos a segmentar, en consecuencia no había problemas de saturación en la imagen. De esta

manera, el proceso de segmentación que se realiza antes de comenzar el reconocimiento de las mitosis dispersas es:

1. Afinar la imagen original obteniendo la imagen P'
2. Dilatar P' dos veces, obteniendo la imagen P''
3. Segmentar sobre la imagen P''
4. Recuperar la imagen P' y sobreponer los contornos sobre ella

IV.4. Otras Operaciones

Como se mencionó, la operación de búsqueda de características no se hizo aislada de otros procesos y operaciones. En esta sección se hablará de ellos. Para calcular el índice mitótico, estos procesos tienen igual importancia para el desempeño del sistema que la búsqueda de características de las categorías.

IV.4.1. Escala de Visualización

Una de las primeras acciones fue cambiar el tipo de objetivo usado para observar las imágenes. Manualmente, el reconocimiento se hace por medio de un objetivo de 20 aumentos pero en el desarrollo del sistema se usó un objetivo de 10 aumentos; esto se debe a que el tiempo de barrido que se necesita para recorrer una laminilla aumenta en gran medida al usar en el sistema un objetivo de 20 aumentos.

La determinación del uso del objetivo x10 surgió al momento de realizar barridos de experimentación junto con la búsqueda de parámetros que caracterizarán a los núcleos estimulados. Con esto se observó que el traslado de la platinas era bastante más lento, al usar los desplazamientos definidos con el objetivo de veinte aumentos, haciendo al sistema salirse del tiempo esperado para su ejecución, pero al usar un objetivo de menos aumento la información todavía era obtenible y el barrido era bastante más rápido, pues los desplazamientos definidos para él abarcan una superficie mayor.

Se debe recordar que el sistema posee la opción de cambiar el valor del objetivo que se está usando para calcular los desplazamientos por medio de la función `defobj`, así que no es obligatorio el uso del objetivo de 10 aumentos. Sin embargo toda la búsqueda de parámetros de las mitosis y núcleos se basó en el uso de este aumento para el análisis de las muestras.

IV.4.2. Orden de Búsqueda de Parámetros

Otra acción llevada a cabo fue la de verificar el orden de las operaciones de búsqueda de cada parámetro. Dentro de la implementación de las búsquedas de parámetros es

necesario realizar a su vez un examen de algunos parámetros antes que otros debido al ahorro de tiempo o porque es necesario realizar determinados procesos antes de efectuar la búsqueda.

Dentro de la definición de parámetros se descubrió que para el caso de los núcleos estimulados era necesario buscar primero los parámetros más importantes, área y factor de forma. Si estos eran satisfechos, entonces se buscaría la desviación estándar del nivel de gris dentro del contorno.

En el caso de las mitosis compactas se halló que hay mayor importancia en el orden de búsqueda, pues hay mediciones que pueden ser realizadas antes de la erosión de los contornos y otras que se deben hacer después de la erosión. La búsqueda del área puede ser realizada antes de la erosión de los contornos, pero posteriormente deben ser revisados los parámetros de la siguiente manera: primero se examinan la desviación estándar del nivel de gris y del nivel de gris medio dentro del contorno; si este es satisfecho entonces se revisa si los cambios por línea, los cambios por pixeles y el contraste interno cumplen las condiciones de una mitosis compacta. Si estos parámetros son satisfechos, se lleva a cabo la última revisión, se verifica si la densidad de grises dentro del contorno está dentro de los límites de las mitosis compactas. Una vez que todos los parámetros son satisfechos entonces el contorno es clasificado como mitosis compacta.

En el caso de las mitosis dispersas el orden también es importante. La búsqueda de parámetros y la optimización del código dieron como resultado que la revisión debe ser hecha de la siguiente manera: primero se buscan los parámetros de desviación estándar de nivel de gris, el cual da un primer indicio de la textura del contorno, luego el área, que restringe la clasificación de aquellos contornos demasiado grandes o muy pequeños. Cuando se cumplen estos parámetros se buscan los cambios por línea y por pixeles horizontales y verticales, que son parámetros de textura más precisos. Cuando estos parámetros satisfacen los límites definidos de las mitosis dispersas, se continúa la búsqueda de la relación de pendientes y la distribución de los niveles de grises; si estos últimos parámetros son satisfechos, entonces el contorno se clasifica como mitosis dispersa.

IV.4.3. Orden de Búsqueda de Categorías

Se deben tomar en cuenta otros factores en la búsqueda de objetos, como el orden de búsqueda de cada categoría. Desde el inicio se decidió que las mitosis dispersas serían las últimas en ser buscadas debido a que habría un ahorro en el tiempo de búsqueda (si ya hubieran sido clasificados algunos objetos antes de comenzar la clasificación de las mitosis dispersas). Pero había que definir que buscar primero si las mitosis compactas o los núcleos estimulados.

necesario realizar a su vez un examen de algunos parámetros antes que otros debido al ahorro de tiempo o porque es necesario realizar determinados procesos antes de efectuar la búsqueda.

Dentro de la definición de parámetros se descubrió que para el caso de los núcleos estimulados era necesario buscar primero los parámetros más importantes, área y factor de forma. Si estos eran satisfechos, entonces se buscaría la desviación estándar del nivel de gris dentro del contorno.

En el caso de las mitosis compactas se halló que hay mayor importancia en el orden de búsqueda, pues hay mediciones que pueden ser realizadas antes de la erosión de los contornos y otras que se deben hacer después de la erosión. La búsqueda del área puede ser realizada antes de la erosión de los contornos, pero posteriormente deben ser revisados los parámetros de la siguiente manera: primero se examinan la desviación estándar del nivel de gris y del nivel de gris medio dentro del contorno; si este es satisfecho entonces se revisa si los cambios por línea, los cambios por píxeles y el contraste interno cumplen las condiciones de una mitosis compacta. Si estos parámetros son satisfechos, se lleva a cabo la última revisión, se verifica si la densidad de grises dentro del contorno está dentro de los límites de las mitosis compactas. Una vez que todos los parámetros son satisfechos entonces el contorno es clasificado como mitosis compacta.

En el caso de las mitosis dispersas el orden también es importante. La búsqueda de parámetros y la optimización del código dieron como resultado que la revisión debe ser hecha de la siguiente manera: primero se buscan los parámetros de desviación estándar de nivel de gris, el cual da un primer indicio de la textura del contorno, luego el área, que restringe la clasificación de aquellos contornos demasiado grandes o muy pequeños. Cuando se cumplen estos parámetros se buscan los cambios por línea y por píxeles horizontales y verticales, que son parámetros de textura más precisos. Cuando estos parámetros satisfacen los límites definidos de las mitosis dispersas, se continúa la búsqueda de la relación de pendientes y la distribución de los niveles de grises; si estos últimos parámetros son satisfechos, entonces el contorno se clasifica como mitosis dispersa.

IV.4.3. Orden de Búsqueda de Categorías

Se deben tomar en cuenta otros factores en la búsqueda de objetos, como el orden de búsqueda de cada categoría. Desde el inicio se decidió que las mitosis dispersas serían las últimas en ser buscadas debido a que habría un ahorro en el tiempo de búsqueda (si ya hubieran sido clasificados algunos objetos antes de comenzar la clasificación de las mitosis dispersas). Pero había que definir que buscar primero si las mitosis compactas o los núcleos estimulados.

Inicialmente se realizaba primero la búsqueda de núcleos estimulados, sin embargo, como se mencionó anteriormente, algunas mitosis compactas eran clasificadas como núcleos estimulados, lo cual modifica radicalmente el cálculo del índice mitótico así que se cambió el orden, primero se buscan las mitosis compactas y posteriormente los núcleos estimulados.

La secuencia de búsqueda es la siguiente:

- 1.- Mitosis compactas
- 2.- Núcleos estimulados
- 3.- Mitosis dispersas

Cada operación de búsqueda está definida en las subrutinas llamadas **recCromosomasCerrados**, **recNúcleosEstimulados** y **recCromosomasAbiertos** respectivamente.

Es pertinente explicar brevemente la forma en que se lleva a cabo el proceso en cada rutina. En la búsqueda de las mitosis compactas, **recCromosomasCerrados** (ver figura 4.19), se van etiquetando con un cierto valor los contornos que cumplen con las características de mitosis compactas, dejando los artefactos sin clasificar. En esta rutina es donde se segmenta la imagen para buscar las mitosis compactas y los núcleos estimulados, esto se debe a que en los mismos contornos se buscan ambos tipos de cuerpos. La etiquetación se hace de la siguiente forma: **BIOCOM** puede dar una etiqueta numérica a cada contorno. Pues se cuenta con una estructura dentro del sistema llamada **CATEG[n]**, la cual es un arreglo de 3600 casilleros que guarda la etiqueta numérica del contorno *n*. Al segmentar, **BIOCOM** actualiza cada campo del arreglo con la etiqueta 1 para el número de contornos hallados.

Esta tabla no es modificada sino al segmentar nuevamente o al cambiar directamente la categoría ya sea por medio de una función de **BIOCOM** o al hacer referencia a dicho arreglo.

Por medio de esta tabla se puede manejar la categoría de los contornos y etiquetarlos para saber cuales ya fueron clasificados y cuales no.

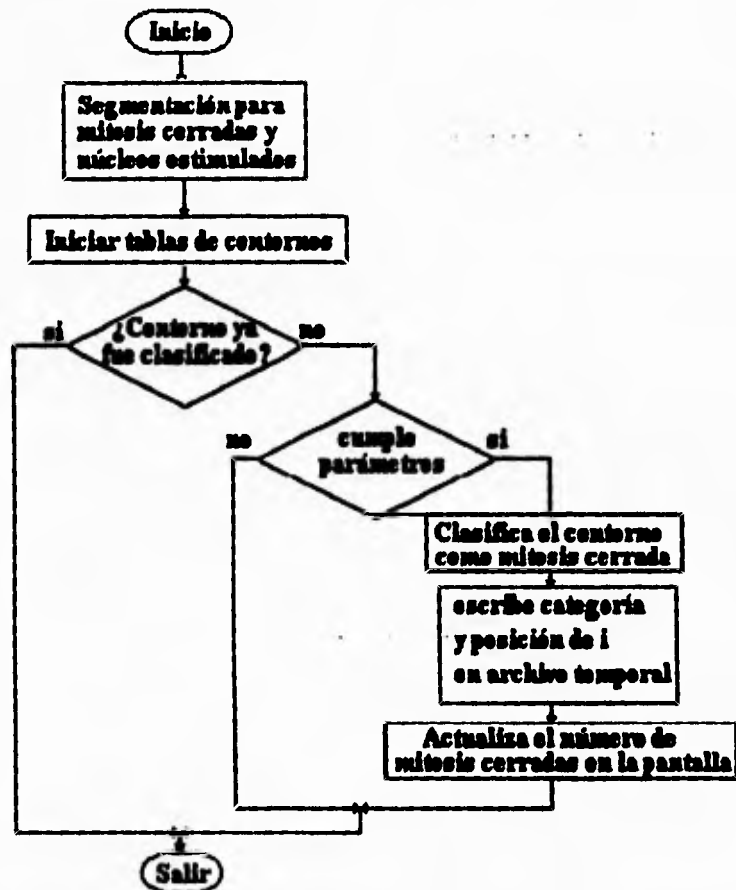


Fig 4.19 Diagrama de bloques para recCromosomasCerrados

Los artefactos tiene cierta etiqueta y los contornos clasificados poseen otra, de ahí que las búsquedas posteriores se restrinjan a únicamente los contornos etiquetados como artefactos. Es de esta forma que los contornos no usados son heredados a la siguiente rutina. Es por ello que solamente se necesita una segmentación para buscar núcleos estimulados y mitosis compactas.

Al buscar los núcleos estimulados en la rutina **recNúcleosEstimulados** (siguiente figura) también se hace una segunda búsqueda de mitosis compactas. Esto es un reminicente de la forma en que estaban las rutinas de búsqueda anterior. Sin embargo dejarla en la practica ha demostrado tener buenos resultados.

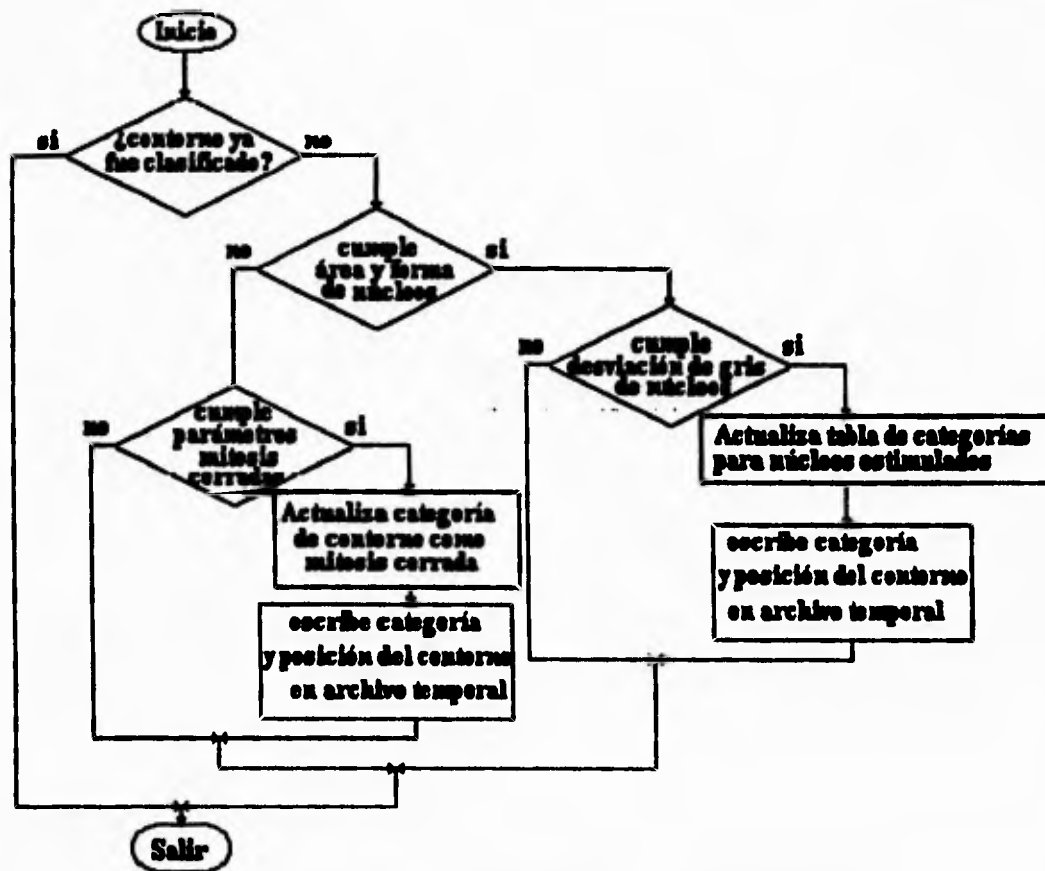


Fig 4.20 Diagrama de bloques para recNucleosEstimulados

En las dos rutinas anteriores la posición y categoría de cada cuerpo clasificado son almacenadas en un archivo temporal, el cual será de gran importancia en la búsqueda de mitosis dispersas.

En la rutina **recCromosomasAbiertos** (figura 4.21) se segmenta nuevamente, eliminando la segmentación anterior y limpiando la tablas de categorías. Después de segmentar se buscan nuevamente aquellos cuerpos que son considerados mitosis dispersas, cuyas posiciones y categorías son almacenadas también en el archivo temporal.

Cuando el orden de búsqueda era:

- 1.- Núcleos estimulados
- 2.- Mitosis compactas
- 3.- Mitosis dispersas

se hacia una búsqueda inicial de mitosis compactas antes de buscar las mitosis dispersas. Esta operación era realizada dilatando la imagen tan solo una vez y segmentando sobre ella. En está se buscaban mitosis compactas que no estuvieran ya clasificadas y sus posiciones y categorías eran almacenadas en el archivo temporal. Esta operación no clasificaba muchas mitosis más, así que al cambiar el orden de la rutinas de búsqueda se retiró esta primera operación de la rutina **recCromosomasAbiertos**.

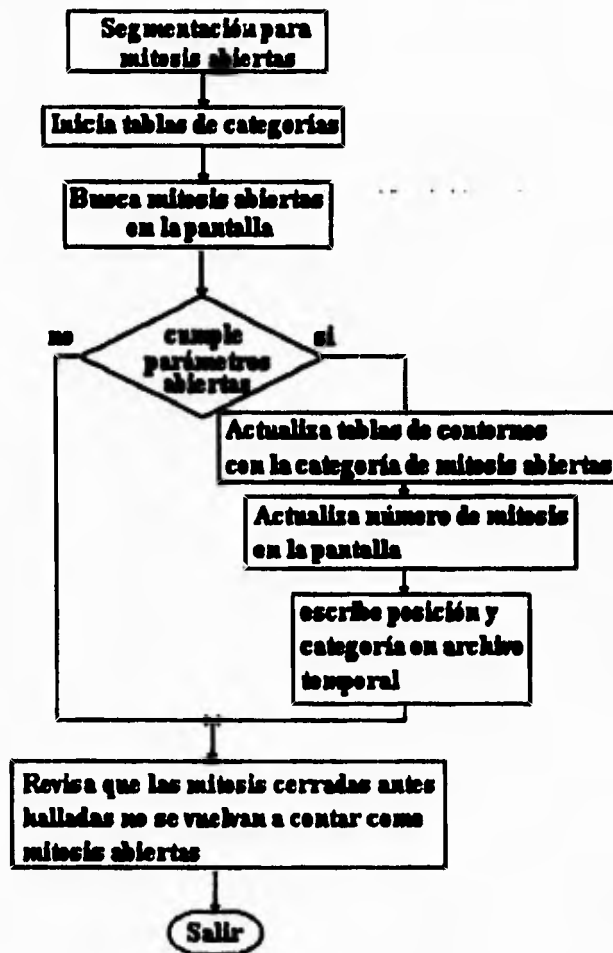


Fig 4.21 Diagrama para la búsqueda de mitosis dispersas

Después de buscar las mitosis dispersas se leen del archivo temporal aquellas posiciones de mitosis compactas para evitar una doble clasificación de las mitosis. Si una mitosis compacta fue clasificada como mitosis dispersa simplemente cambia su categoría a mitosis dispersa dentro del archivo, lo cual no influye en el resultado del índice mitótico.

Posteriormente, ya fuera de las rutinas anteriores, el archivo temporal se vacía en el archivo NÚCLEOS.REC, que guarda la información de la ejecución.

La rutina que realiza el reconocimiento en cada pantalla, usando `recCromosomasCerrados`, `recCromosomasAbiertos` y `recNúcleosEstimulados` y vaciando el archivo temporal al archivo NÚCLEOS.REC, se llama `accion_mito` (figura 4.22).

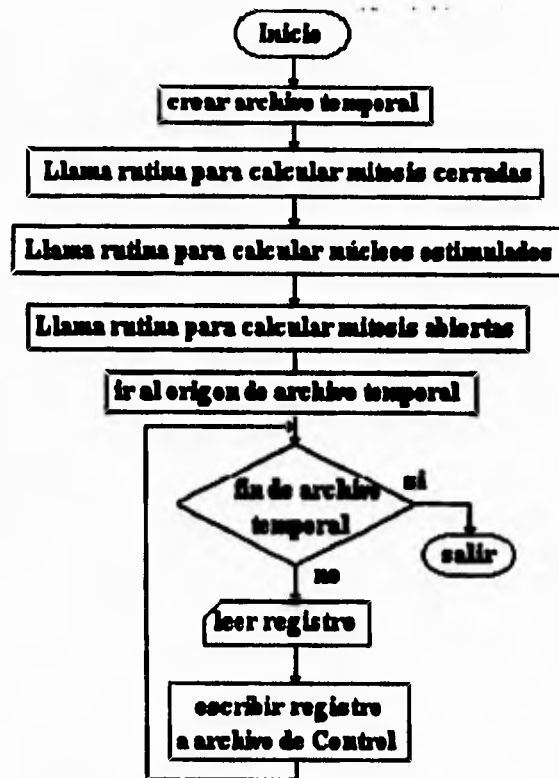


Fig 4.22 Diagrama de la implementación de accion_mito

IV.4.4. Problemas de Segmentación

Durante las pruebas de reconocimiento surgió un nuevo problema debido a que las laminillas poseían una preparación diferente a la que tenían las laminillas anteriores. En esta nueva preparación, los objetos eran muy tenues lo que hacía que el método para hallar el nivel de gris para segmentar extraviara un gran número de cuerpos de interés.

Se presentaba el problema de encontrar un método que permitiera el aumento de contraste en las imágenes pero que no saturara las tablas de contornos. Obviamente no se podía afinar la imagen antes de segmentar pues había demostrado que no era eficiente al provocar la saturación de manera frecuente.

Ante este problema se decidió hacer uso de las LUT's de despliegue, las cuales permiten tener una imagen modificada con una función de transferencia de forma no destructiva. (ver capítulo 1).

Sin embargo, era necesario definir la función de transferencia que modificaría a la imagen. La forma que debería tener dicha función de transferencia se dibujó en la pantalla, gracias a una opción que brinda el mismo sistema que aplica la curva dibujada en la pantalla a las LUT's de despliegue. La figura siguiente muestra esta operación.



a)



b)

Fig. 4.23. Proceso de aplicación manual de la LUT. a) Imagen original.
b) Resultado.

La curva tiene esa forma debido a que es necesario resaltar los niveles de gris oscuros que representan objetos, y dejar sin modificación a los niveles claros, el fondo. Es por ello que la curva debía crecer súbitamente.

Haciendo una interpolación de la curva obtenida se vió que era de forma cuadrática o cuarta y que, dentro de la pantalla, finalizaba su crecimiento en un punto específico determinado por la moda del histograma y comenzaba en el origen, valor cero.

La curva está parametrizada, es decir que la parte superior de la pantalla representa el valor 1 en las ordenadas mientras que la inferior representa 0 (Fig. 4.23). Teniendo esta información se hizo lo siguiente:

de la ecuación: $\alpha x^4 = 1$ (1)
 $\alpha x^2 = 1$

que provienen de las ecuaciones $ax^4 + bx^3 + cx^2 + dx + e = 1$ que afectarán a las tablas de asignación (LUT's), para cada valor de gris de entrada (x) se modifica para dar un valor de gris de salida f(x). Estas funciones son calculadas en terminos del porcentaje de blanco con una precisión de 1/255 (ocho bits) y una representación en número entero (ver figura siguiente).

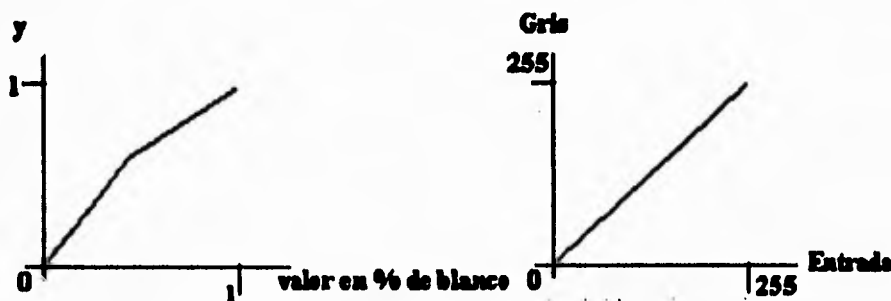


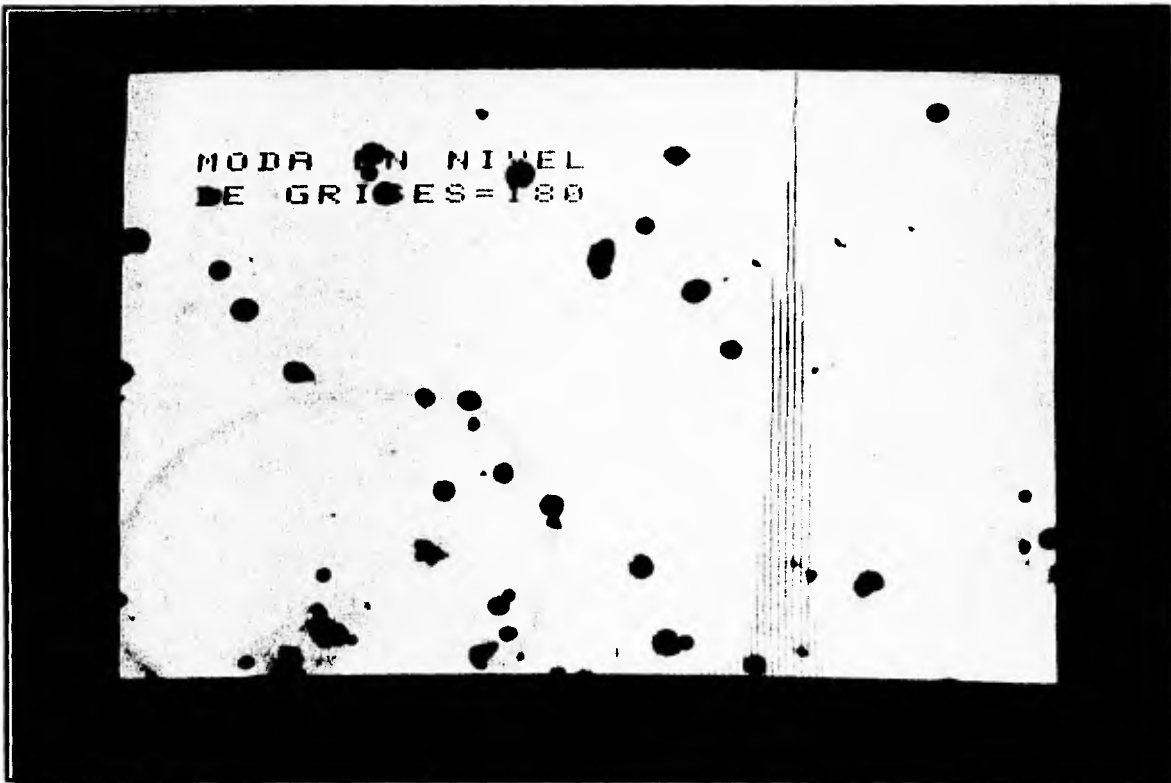
Fig. 4.24 Comportamiento de las tablas de asignación.

De las ecuaciones (1) se debe encontrar el valor de α tal que

$$\alpha = \frac{1}{x^4} = \left(\frac{1}{x}\right)^4$$

También se debe mencionar que la curva está definida en el intervalo cerrado $[0, 1]$ así que el valor 256 del histograma equivale a 1 en el eje X; esto significa que cuando la moda equivale al número máximo de grises, 256, entonces $X=1$. Con esta relación, α queda como sigue:

$$\alpha = \left(\frac{1}{\left(\frac{MODA}{MAX_GRIS} \right)} \right)^4 = \left(\frac{MAX_GRIS}{MODA} \right)^4$$

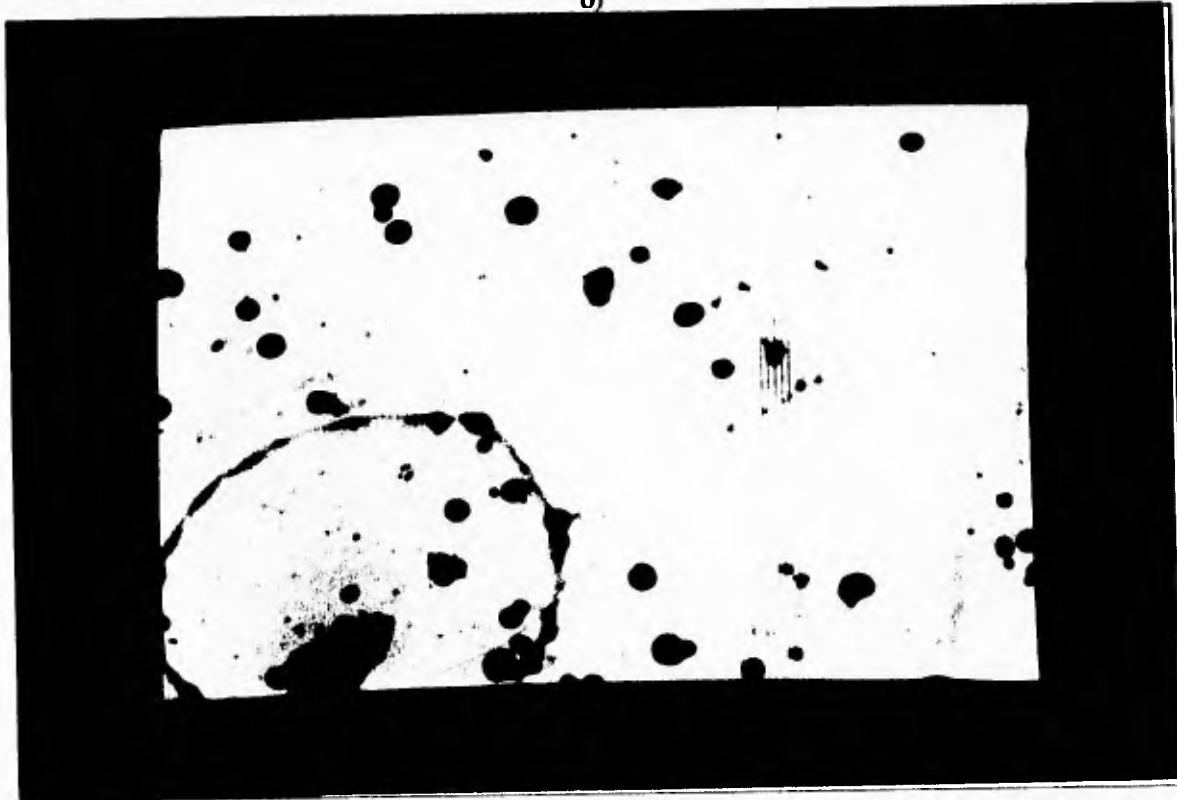


a)

Fig. 4.25. Proceso de aplicación de las LUT's. a) Imagen original.
 b) Cálculo del coeficiente α . c) Imagen de curva e histograma con las LUT's aplicadas.
 d) Resultado de aplicación de LUT's.



b)



c)

Fig. 4.25. Proceso de aplicación de las LUT's. a) Imagen original.
b) Cálculo del coeficiente α . c) Imagen de curva e histograma con las LUT's aplicadas.
d) Resultado de aplicación de LUT's.

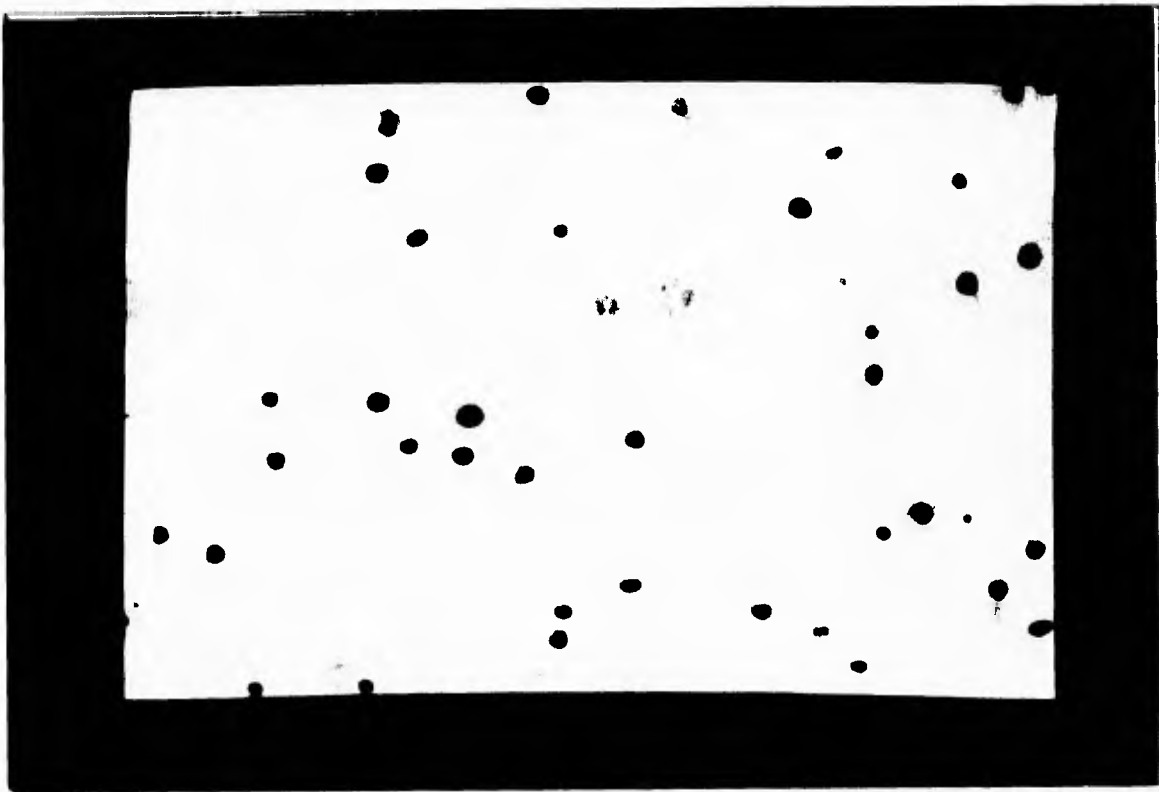


d)
Fig. 4.25. Proceso de aplicación de las LUT's. a) Imagen original.
b) Cálculo del coeficiente α . c) Imagen de curva e histograma con las LUT's aplicadas.
d) Resultado de aplicación de LUT's.

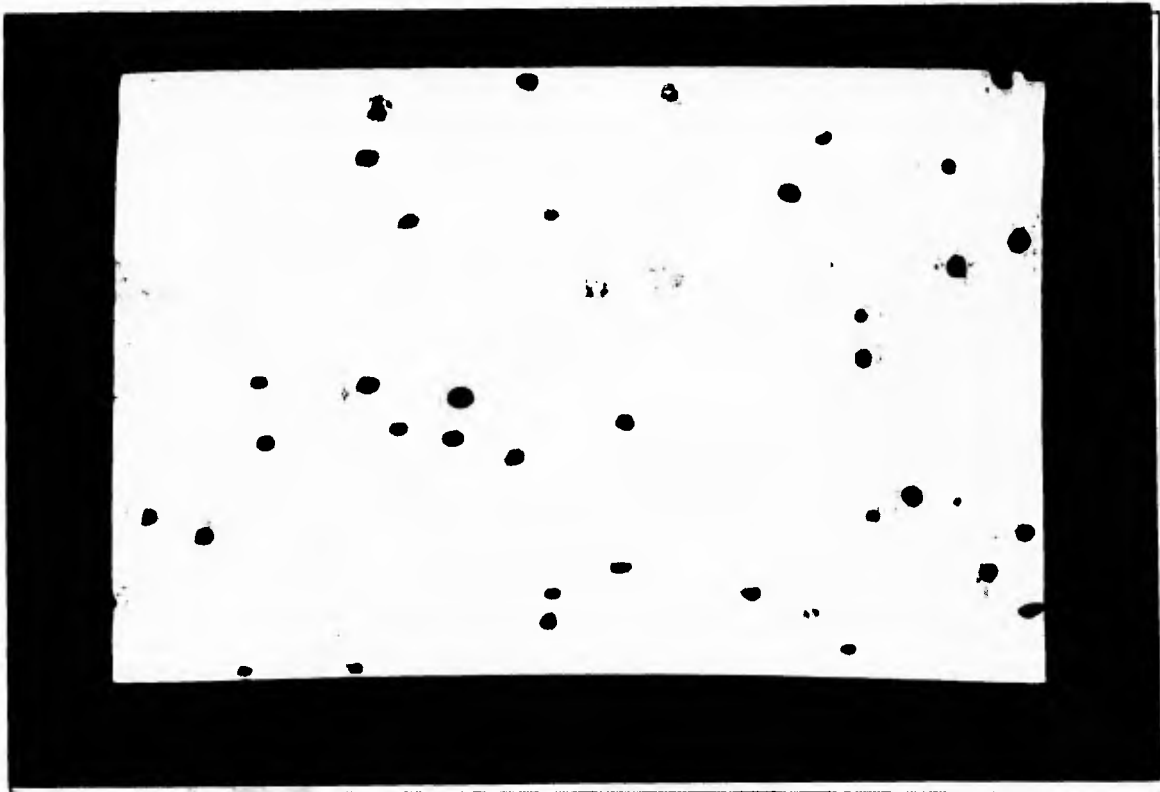
Finalmente había que decidir qué curvas usar, si una cuadrática o una de cuarto orden. Al realizar las pruebas se detectó que la curva que más servía para aumentar el contraste era la de cuarto orden, pero se deja la opción de la curva de segundo orden por si es necesario usarla en el futuro.

Después de usar la curva de transferencia se requiere hacerla vigente en el *buffer* de la imagen; es decir que sea aplicada a la imagen, pues antes solamente se vió su posible resultado sin modificar el *buffer* de almacenamiento de la imagen. La secuencia de la operación se puede ver en la figura 4.25.

Es necesario realizar este proceso antes de segmentar y posteriormente llevar a cabo el proceso de segmentación sobre la imagen modificada, para obtener los resultados deseados. Posteriormente las LUT's son retornadas a su estado original para realizar las operaciones de búsqueda. La imagen original y posterior a la aplicación de las LUT's se muestra en la siguiente figura.



a)



b)

Fig. 4.26. Segmentacion modificando LUT's a) Imagen original
b) Resultado

Como se puede ver, cada vez que se segmenta, en el proceso del cálculo del índice mitótico (**recCromosomasCerrados** y **recCromosomasAbiertos**) se utiliza este método por medio de las subrutinas **FMODA**, que regresa el valor de la moda, y **FUNCIO**, que aplica la función de transferencia a las LUT's y las regresa a su estado inicial.

IV.4.5. Entrenamiento

Durante la explicación anterior se mencionó qué parámetros caracterizan a los objetos que se buscan y el orden de búsqueda, pero como el valor de estos parámetros no está fijo en el sistema en algún momento el usuario debe definir los valores de ellos. Esta tarea podría ser titánica si el sistema no contara con una opción de entrenamiento en donde, por un lado, se puede elegir un archivo conteniendo los valores de los parámetros y, por el otro, la posibilidad de crear uno de estos archivos de manera interactiva.

La primera de las opciones se realiza por medio de la subrutina **acttr** que busca archivos con la terminación **ENT**. Esta subrutina permite elegir por medio del ratón alguno de los archivos con terminación **ENT** que existan en el directorio de trabajo.

Por medio del uso de listas se crea un censo de los archivos **.ENT** existentes y se despliegan en la pantalla para que el usuario pueda seleccionar alguno de ellos, o en su caso ninguno. Cuando no se selecciona ninguno el sistema guarda como activo el último que estuvo activo; si se desactivan todos los entrenamientos entonces la operación de cálculo no puede ser llevada a cabo.

La operación de definición de un archivo nuevo se lleva a cabo en la rutina **nuetr**. La filosofía de esta rutina es que genere un archivo conteniendo los valores de cada uno de los parámetros de los objetos a clasificar por medio de una muestra representativa, haciendo que el usuario indique en la pantalla que objetos son núcleos y cuáles son mitosis.

El sistema tiene la capacidad de ir calculando cada uno de los parámetros de los objetos seleccionados en cada pantalla y acumularlos en variables para cuando se termine de ejecutar la rutina **nuetr**.

El funcionamiento de la subrutina se puede esquematizar en la figura 4.27.

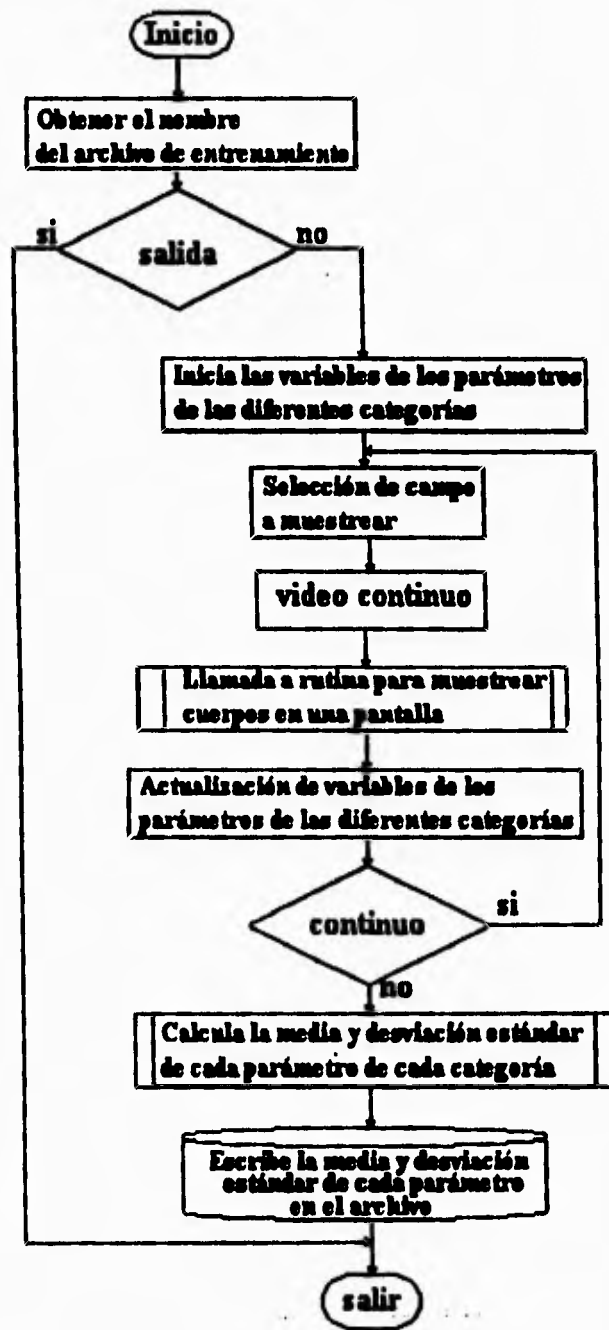


Fig 4.27. Esquema del funcionamiento de la rutina nuetr.

La función que calcula la media y desviación estándar de cada característica de cada

categoría se lleva a cabo por medio de la fórmula:

$$\text{desviación} = \sqrt{\left(\frac{\sum X^2}{n} - \left(\frac{\sum X}{n}\right)^2\right)}$$

$$\text{media} = \frac{\sum X}{n}$$

La subrutina que selecciona y acumula los valores de cada parámetro que caracteriza a cada cuerpo en un campo de muestreo es `calc_cont` y realiza las siguientes operaciones:

1. Permite seleccionar contornos que representan a cada categoría.
2. Calcula los valores de cada parámetro que caracteriza a cada categoría.
3. Pasa cada valor a la rutina padre.

y cuyo funcionamiento se muestra en la siguiente figura.



Fig. 4.28. Esquema del funcionamiento de la rutina calc_cont.

En el proceso de selección de los objetos, cuando se eligen mitosis compactas y núcleos, simplemente se marcan con distintas etiquetas para que al terminar la selección de objetos el cálculo de sus características se haga directamente y se almacenen en variables que serán regresadas al salir de la rutina.

En el caso de las mitosis dispersas se debe realizar un proceso distinto, ya que la imagen con la que se trabaja y se selecciona a los objetos no es la que se usa para la detección de mitosis dispersas. Así que como se puede ver, se realizan dos tipos diferentes de segmentaciones y, por ende se renuevan las tablas de contornos. Esto lleva a que se deba guardar de alguna manera la posición de los cromosomas que conforman a la mitosis dispersa para que luego se busque en la segmentación sobre la imagen, doblemente dilatada, aquel contorno que contenga a todos los cromosomas que se seleccionaron, o a la mayoría. Para esto se usa una elipse que encierra a los cromosomas y cuyos centros de gravedad, radio menor y radio mayor, son guardados en una lista ligada en forma de pila que ayuda a saber en que zona de la pantalla se halla la mitosis seleccionada.

IV.4.6. Búsqueda de límites para los parámetros

Por último, hay otro proceso que se llevó a cabo paralelamente a la investigación de las características de cada categoría: la búsqueda de los límites precisos de cada uno de ellos para hacer el filtrado de los contornos.

La desviación estándar expresa algo de la variabilidad de una muestra con respecto al valor medio. Por lo tanto, la probabilidad de que las variables aleatorias, que representan a cada una de las características de los objetos a buscar, asuman un valor dentro de cierto intervalo con respecto a la media tiene que ver con el tamaño de la desviación estándar de la muestra.

La estimación de la probabilidad de que una variable aleatoria tome un valor dentro de k desviaciones estándares a partir de su media para cualquier número real k está dada por el teorema de Chebyshev que dice:

$$P(\mu - k\sigma < X < \mu + k\sigma) \geq 1 - \frac{1}{k^2}$$

donde μ es la media y σ es la desviación estándar. Si se toman dos desviaciones estándares, entonces, el resultado es 75% de probabilidad de que las observaciones estén dentro del intervalo $\mu \pm 2\sigma$. Se debe recordar que el teorema de Chebyshev, por un lado, se aplica para cualquier distribución de observaciones debido a que los resultados son generalmente inseguros y, por el otro, que esto representa tan solo una cota inferior lo que significa que nunca se sabe de cuanto será la probabilidad máxima de que la variable aleatoria se

encuentre dentro de los límites $\mu \pm k\sigma$, esta probabilidad solamente será conocida cuando se conozca con exactitud la distribución de probabilidad.

La forma en que se hace el filtrado de cada uno de los parámetros que caracterizan a los objetos, es tomando en cuenta el teorema de Chebyshev para restringir la entrada de datos (figura 4.29). Es decir, se usan k desviaciones estándares para tener menor probabilidad de que los parámetros de las características de los objetos tomen los valores que caracterizan a los artefactos, pero que sí tomen los valores que caracterizan a los objetos.

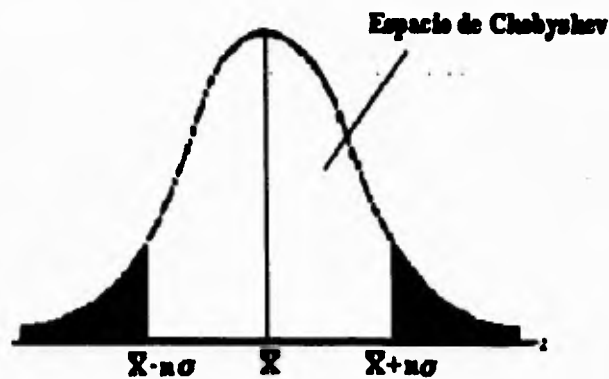


Fig. 4.29 idea de la inclusión usando Chebyshev

Sin embargo el método anterior únicamente logra definir la restricción para un solo parámetro, pero se necesita restringir a los diferentes parámetros combinados. Esto quiere decir que se requiere de lograr una intersección de la curva de distribución de cada característica para formar el filtro de cada objeto, como se muestra en la figura a continuación.

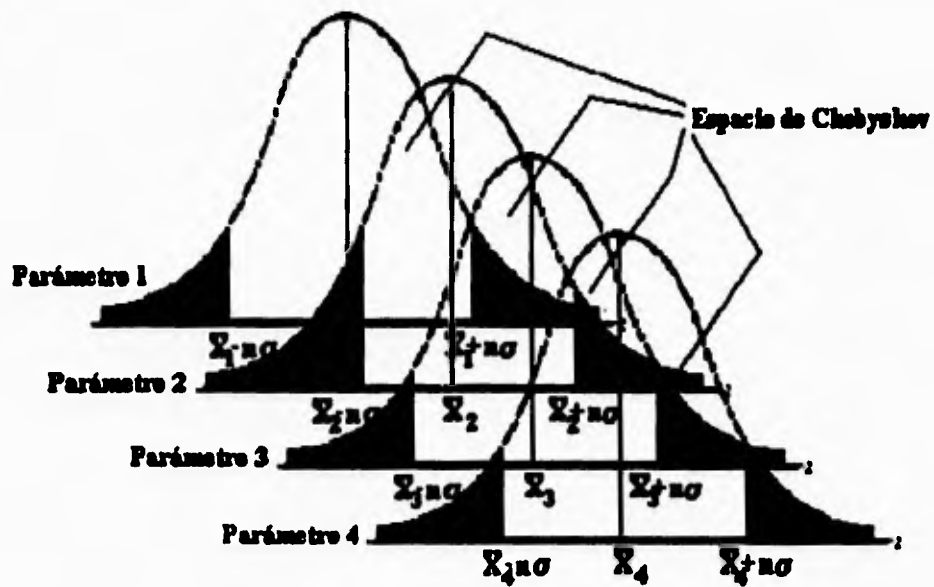
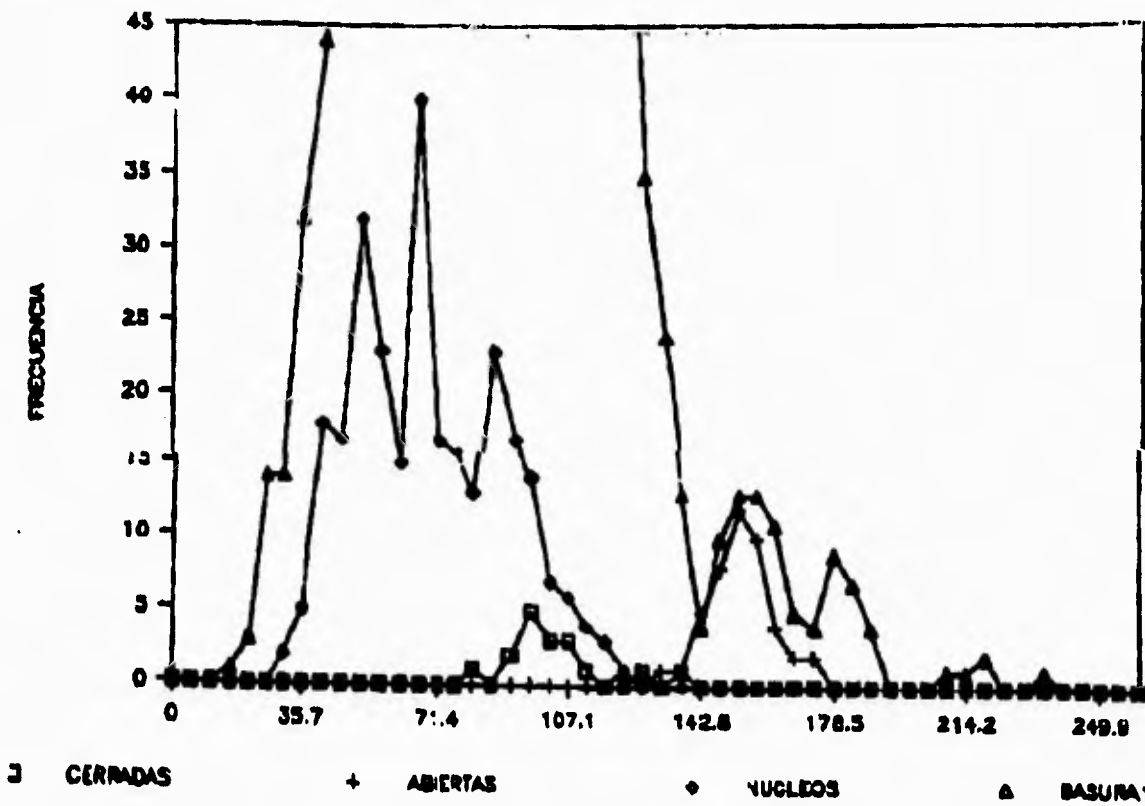


Fig. 4.30 idea de la intersección de la curvas de distribución

Como no se conoce la distribución exacta de cada una de las características de los objetos, se hizo un cálculo de k de forma experimental durante la búsqueda de parámetros, y se optimizaron los valores de k durante la etapa de implementación final, guiándose por las siguientes gráficas de las muestras de cada característica:

NGRIS



a)

Fig. 4.31. Esquema del comportamiento de las características para las diferentes categorías a) Nivel de gris.
b) Factor de forma.

FORMA

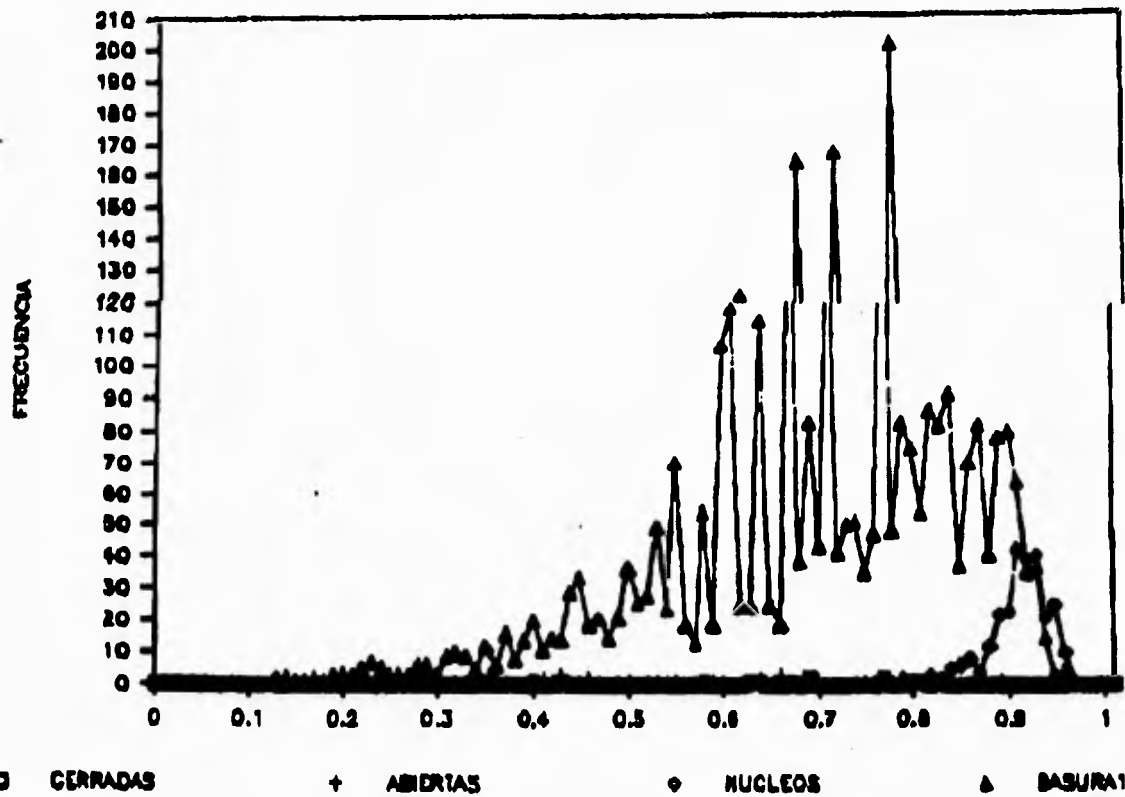


Fig. 4.31. Esquema del comportamiento de las características para las diferentes categorías a) Nivel de gris.
b) Factor de forma.

Como puede observarse es importante lograr el punto óptimo de k para cada curva, debido a que de otra manera el reconocimiento se estará realizando mal y por consiguiente se tendrá un mal cálculo del índice mitótico.

Aquí es importante hacer notar que todavía hace falta realizar un mayor análisis sobre cómo realizar la intersección de las curvas de distribución, de forma tal que se encuentre el punto óptimo de cada valor k para cada característica. Las diferentes formas de hacer este análisis se mencionarán posteriormente en el capítulo V.

IV.5. Rutina de cálculo del Índice Mitótico.

El cálculo del índice mitótico en cada una de las laminillas se lleva a cabo en la rutina mit que se puede ver esquematizada en las siguientes figuras.

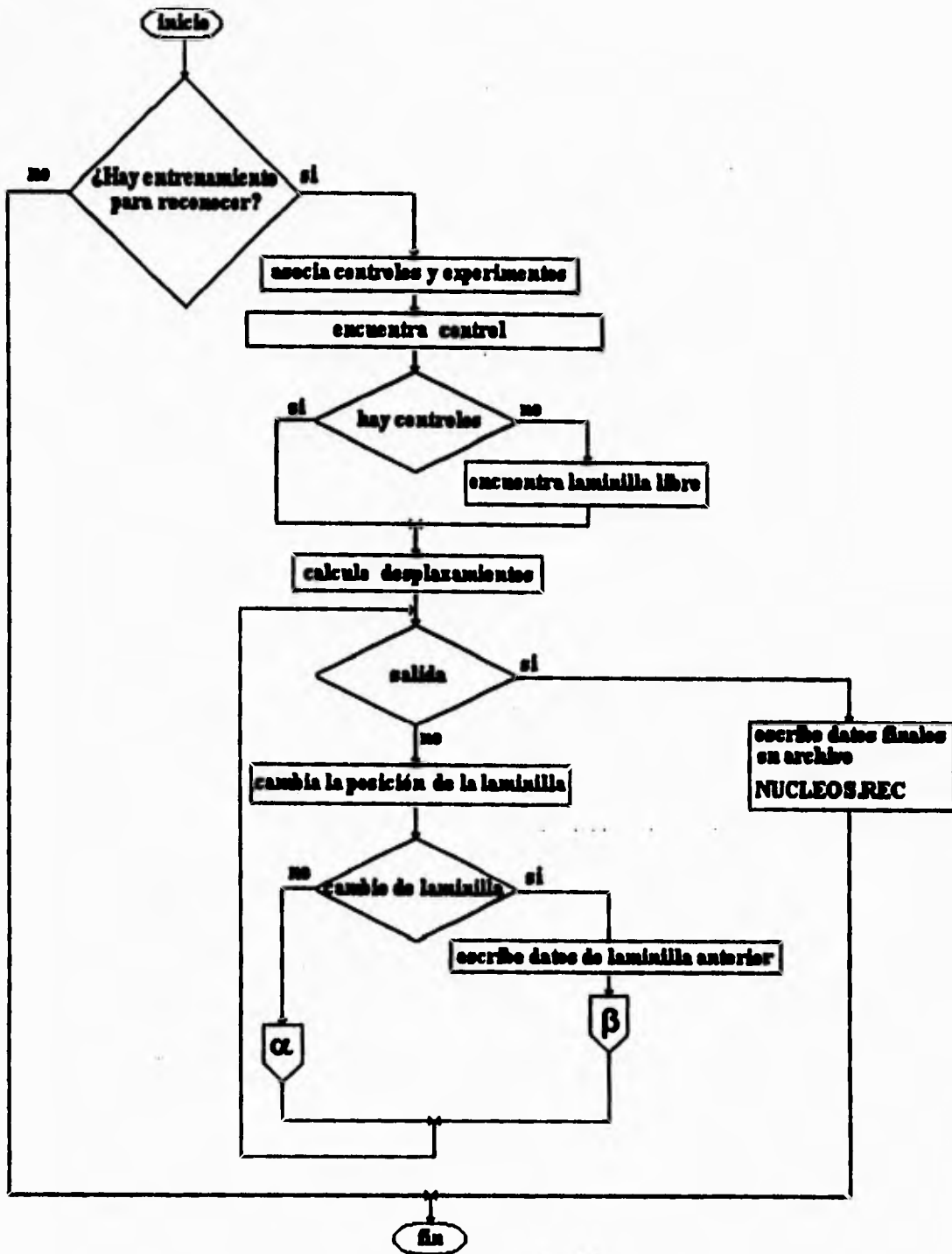


Fig 4.32 Diagrama de flujo principal para la rutina mit.

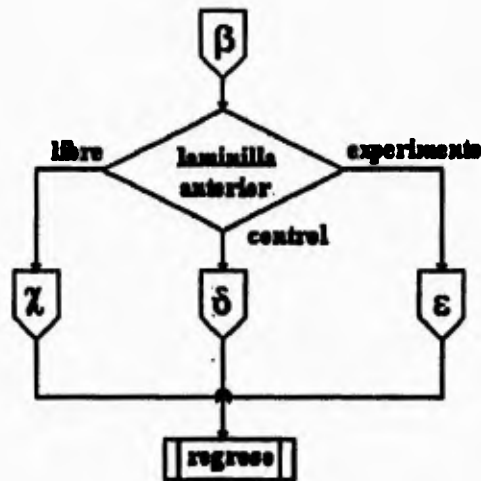


Fig. 4.33 Continuación del diagram de flujo de la figura 4.32.
Sección de control para la búsqueda de la laminilla.

En esta sección se verifica de que tipo era la laminilla anterior para saber que laminilla debe ser buscada posteriormente.

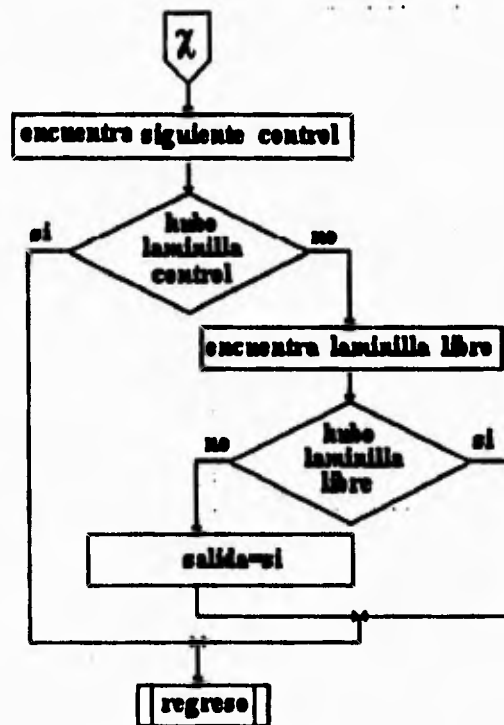


Fig. 4.34 Continuación del diagrama de flujo de la figura 4.33.
Búsqueda de laminillas tipo control.

La sección anterior del diagrama de flujo permite la búsqueda de laminillas que son del tipo *control*. En dado caso que no haya laminilla *control* entonces busca una laminilla tipo *libre*.

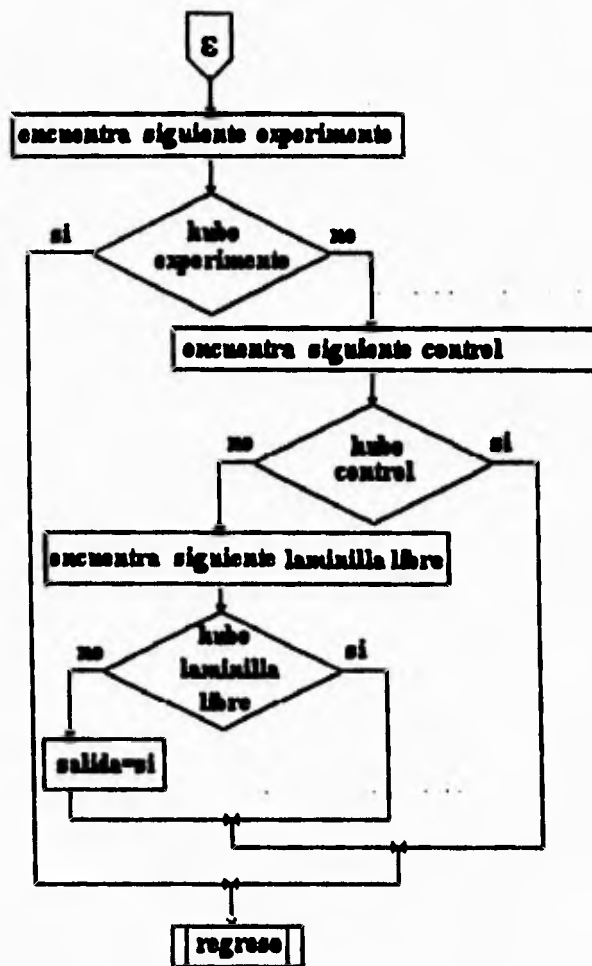


Fig. 4.35 Continuación del diagrama de flujo de la figura 4.33.
Búsqueda de laminillas tipo experimento.

La sección anterior del diagrama de flujo permite la búsqueda de laminillas que son del tipo *experimento*. En dado caso que no haya laminilla *experimento* entonces busca una laminilla tipo *control* o tipo *libre*.

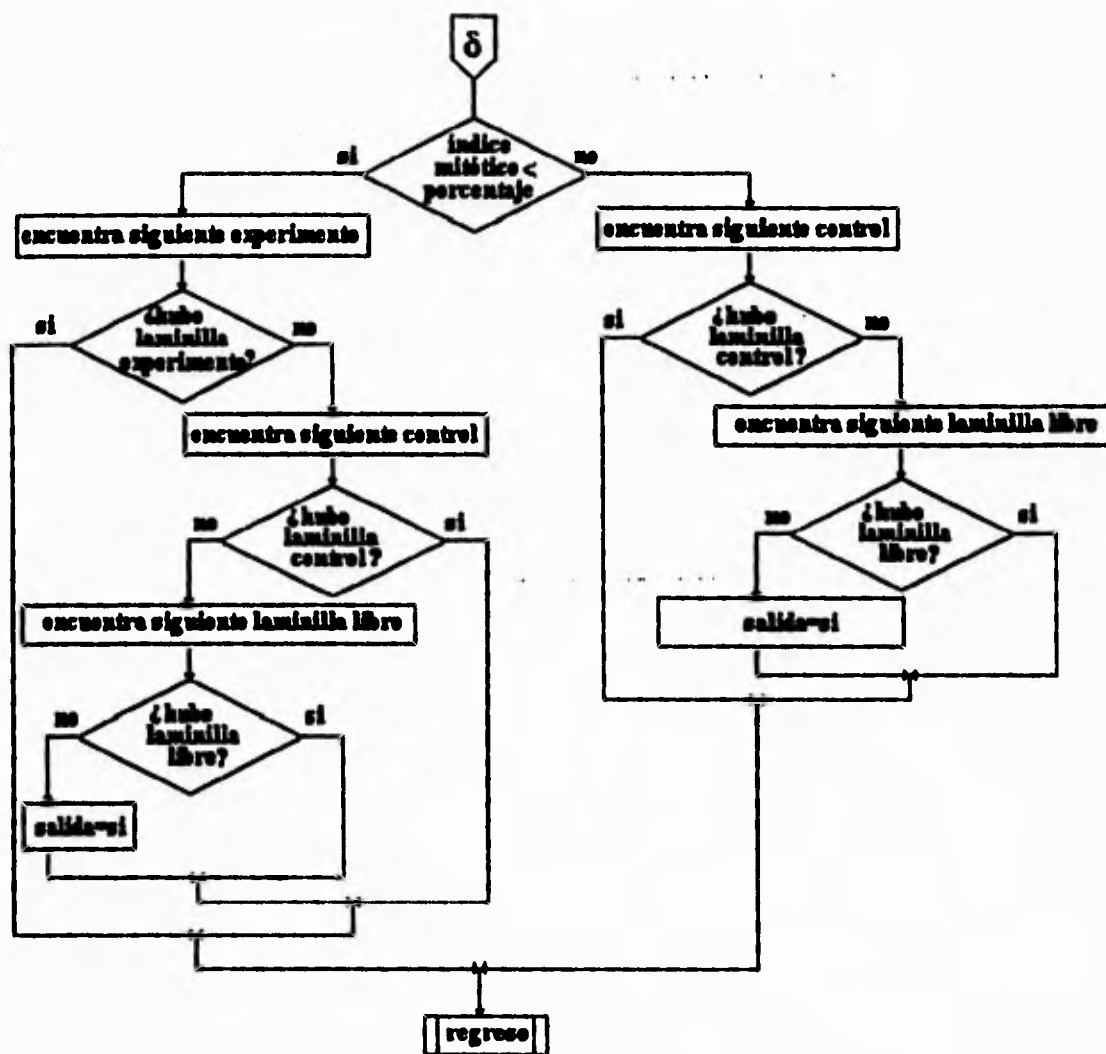


Fig. 4.36 Continuación del diagrama de flujo de la figura 4.33.
Verificación de que se cumple el índice mitótico mínimo para las laminillas tipo control.

En la rutina mit, al terminar de revisar una laminilla tipo *control* entonces se revisa si esta laminilla cumple con el índice mitótico mínimo, si la laminilla *control* iguala o sobrepasa dicho índice mitótico entonces es necesario buscar una laminilla *experimento* que este asociada al *control*, por el contrario se buscará otra laminilla *control* o en su defecto una laminilla *libre*.

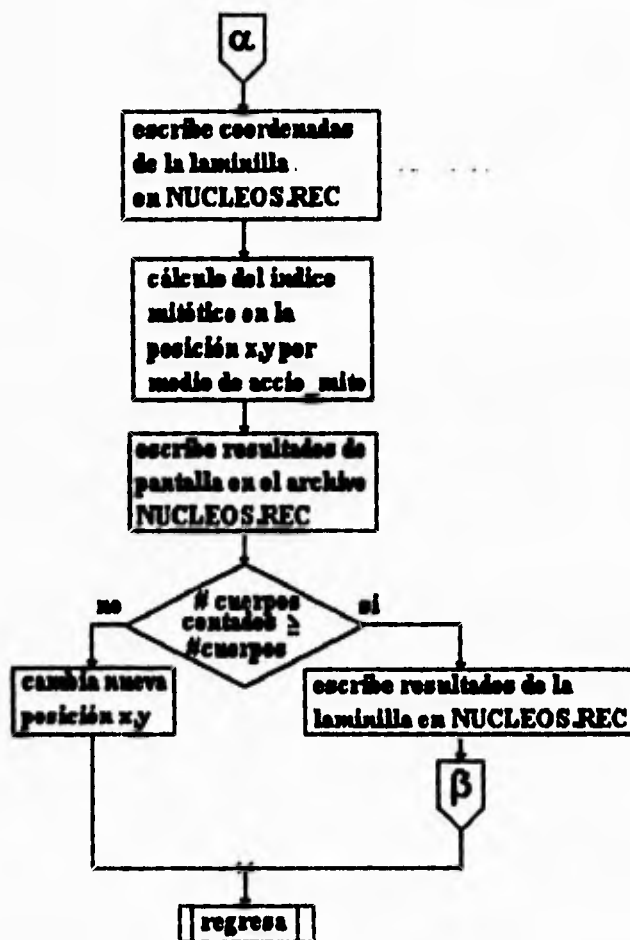


Fig. 4.37 Continuación del diagrama de flujo de la figura 4.32. Análisis en cada campo del barrido.

Por último, se muestra la parte central del proceso del cálculo del índice mitótico y es el reconocimiento y conteo de los diferentes cuerpos en cada campo. Si se alcanza el número máximo de cuerpos a contar entonces se hace la búsqueda de la nueva laminilla.

En la estructura de la rutina `mit` también es posible observar que existen las variables `entrena_existe`, `objetivo_definido` y `overlap_existe`; `entrena_existe` define la existencia de un entrenamiento para el sistema de reconocimiento y que está activo para comenzar el cálculo del índice mitótico, `objetivo_definido` permite saber si fue seleccionado un desplazamiento para las platinas según el objetivo de observación y por último `overlap_existe` tan solo se refiere a si se definió una cierta sobreposición de las tomas. Si estas banderas no están encendidas entonces no se puede realizar la operación de la rutina `mit`.

Otra cosa notable del esquema es la forma en que se lleva control de las laminillas para cumplir las especificaciones de los biólogos. Las rutinas reales de control fueron

mencionadas antes, pero es esencial tener en cuenta que ocupan una parte importante del esquema.

IV.6. Rutinas Relacionadas

Existen dos rutinas relacionadas con aquella que permite el análisis del índice mitótico, estas son: *leecmd* y *mitcmd*. La primera rutina permite la combinación del movimiento de las platinas, por medio de barridos, y aprovechar uno de los atributos de BIOCUM al poder ejecutar un archivo de comandos de línea. En el caso de la rutina *mitcmd* se quiso aprovechar que se podía hacer la ejecución de un archivo de comandos para que, en el caso de que un usuario lo deseara, se pudieran hacer procesos extra antes de realizar el análisis del índice mitótico de un campo, después de haber movido las platinas a un campo determinado.

Para poder llevar a cabo la implementación de estas rutinas fue necesario ver cómo se realiza el proceso para ejecutar un archivo de comandos. El proceso se realiza por medio de un *buffer* físico de control, el apuntador de la posición del mismo *buffer* y una variable de control del tipo de ejecución del sistema, que permite saber el tipo de ejecución que se está llevando a cabo. Cuando se está ejecutando el archivo de comandos la variable de control del tipo de ejecución es igual a uno y cuando el sistema se encuentra en operación normal la variable es igual a cero. Cuando se ha terminado de ejecutar el archivo de comandos el apuntador del *buffer* de control es igual a uno y la primera posición del *buffer* físico de control está ocupada por el carácter cero.

Cuando se pide que un comando sea ejecutado se llena un *buffer* lógico de comandos con el nombre del comando y se decodifica en cada módulo conforme va circulando en el ciclo del programa (ver figura 2.2). Si en ningún módulo encontró concordancia, entonces el sistema marca un error y espera por el siguiente comando a decodificar en el módulo de decodificación de comandos. Si, por el contrario, el *buffer* de comandos halló concordancia en algún módulo, entonces el comando se ejecuta, el sistema enciende algunas banderas y regresa al módulo de decodificación de comandos para esperar el siguiente comando.

Cuando se ejecutan archivos de comandos el nombre del archivo de comandos es vaciado en el *buffer* físico y cada comando del archivo es descargado en el *buffer* lógico. Durante la ejecución de un archivo de comandos también se tiene encendida la bandera de control del tipo de ejecución con el valor de uno y así el sistema sabe que se está ejecutando un archivo de comandos.

Para permitir la implementación de las nuevas rutinas se incrustaron algunas banderas extras: **BAN1**, **BAN2** y **AC**¹. La primera se usa cuando se realiza la rutina *leecmd*, la segunda cuando se realiza la rutina *mitcmd* y la bandera **AC** sirve en *mitcmd* para saber cuando llevar a cabo la ejecución del archivo de comandos y cuando hacer las operaciones de análisis del índice mitótico para un campo. Estas variables toman su valor inicial en la rutina *inicia*.

Al usar un *buffer* físico de control, el apuntador de la posición del *buffer* y la variable de control del tipo de ejecución del sistema más las banderas **BAN1**, **BAN2** y **AC** se puede saber cuándo dejar que se decodifique normalmente de un comando o cuándo hacer llamadas a las rutinas *leecmd* o *mitcmd*. Esta implementación se puede ver en los archivos *MITCOM.FOR* y *MITCOM2.FOR*.

La rutina *MITCOM.FOR* contiene al decodificador del módulo de mitosis en el cual se encuentra la llamada a la rutina que hace el cálculo del índice mitótico y sus rutinas asociadas, y la rutina *MITCOM2.FOR* contiene al decodificador específico para las rutinas de ejecución dentro de las cuales se hallan *leecmd* y *mitcmd*.

La idea dentro de la rutina *MITCOM* (figura 4.38) es básicamente la de decodificar normalmente los comandos de las rutinas pertenecientes al módulo del índice mitótico, sin embargo también verifica si se está ejecutando alguna de las rutinas *leecmd* o *mitcmd*.

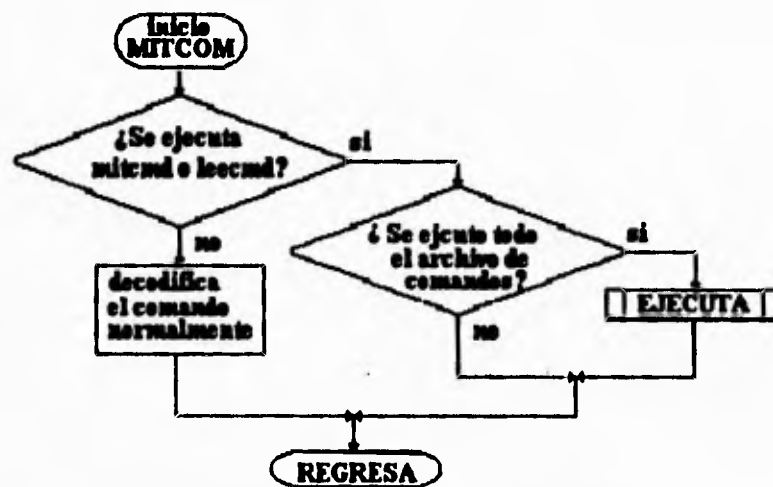


Fig. 4.38 Esquema de funcionamiento de MITCOM.

y la idea de *EJECUTA* (figura 4.39), que es la subrutina que decodifica los comandos *mitcmd*, *mit* y *leecmd*, es saber cuando llamar a cada una de las rutinas anteriores.

¹Para ver cómo se realiza la interfaz entre C y FORTRAN se debe consultar el apéndice G

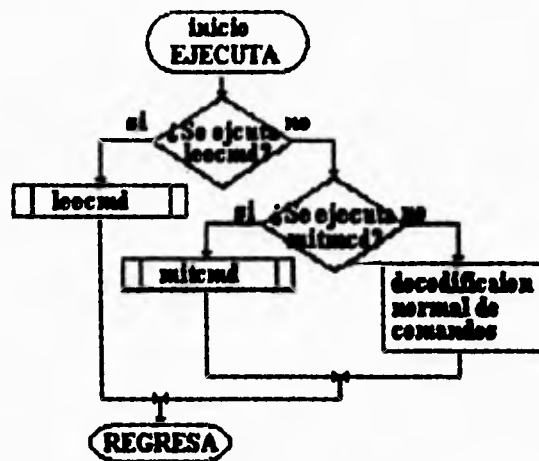


Fig. 4.39 Esquema de funcionamiento de EJECTA.

IV.6.1. Rutina leecmd

La rutina leecmd (figura 4.40) funciona de la manera siguiente:

1. Mueve las platinas a la posición correcta.
2. Ejecuta el archivo de comandos.

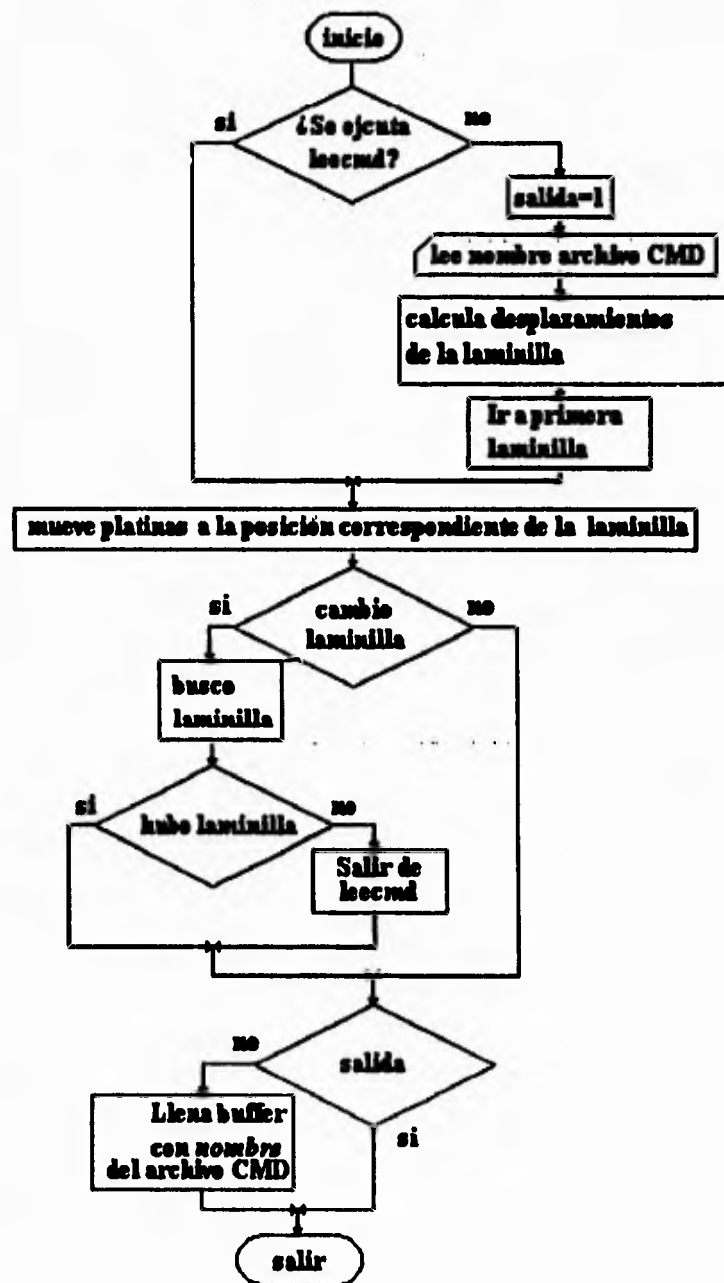


Fig. 4.40 Esquema del funcionamiento de lcccmd.

Esta rutina es un poco diferente a mit y mitcmd pues no lleva un archivo que registre su accionar.

IV.6.1.1. Ejecución del Archivo de Comandos

Para hacer que el sistema lea el archivo de comandos es suficiente con llenar el *buffer* físico con el nombre del archivo antecedido por una arroba: @nombre. La rutina que decodifica todos los comandos de BIOCUM al encontrar una arroba como carácter de inicio sabe que debe ejecutar un archivo de comandos.

IV.6.1.2. Movimiento de las platinas para leecmd

El movimiento de las platinas para leecmd es un poco distinto pues aquí no hay controles ni experimentos, solamente laminillas libres que pueden ser examinadas en cierto orden. Este orden se determina por medio de la rutina orden1 que hace uso de la variable global controles. La forma en que la utiliza es un poco distinta a la manera en que se usa al tener controles y experimentos, como se explicó antes. En este caso lo que se almacena en la estructura es el orden en que serán revisados y la rutina findnext será la encargada de regresar el número de laminilla que debe ser examinada a continuación. Esto se hace hallando el número con menor orden pero mayor a cero, dentro de cada uno de las localidades controles[], pues cuando se ha examinado una laminilla *k* entonces controles[*k*] será igual a cero.

IV.6.2. Rutina mitcmd

Queda por mencionar la rutina mitcmd. Esta rutina es un poco más complicada en su funcionamiento pues se planeó para que realizara los movimientos de las platinas que se necesitan usando controles y experimentos asociados, la ejecución de un archivo de comandos y el proceso del índice mitótico. El orden de cada operación es el que se mencionó antes:

1. Mover platinas
2. Ejecutar todo el archivo de comandos
3. Calcular el índice mitótico en el campo visual.

El esquema de la rutina es el siguiente:

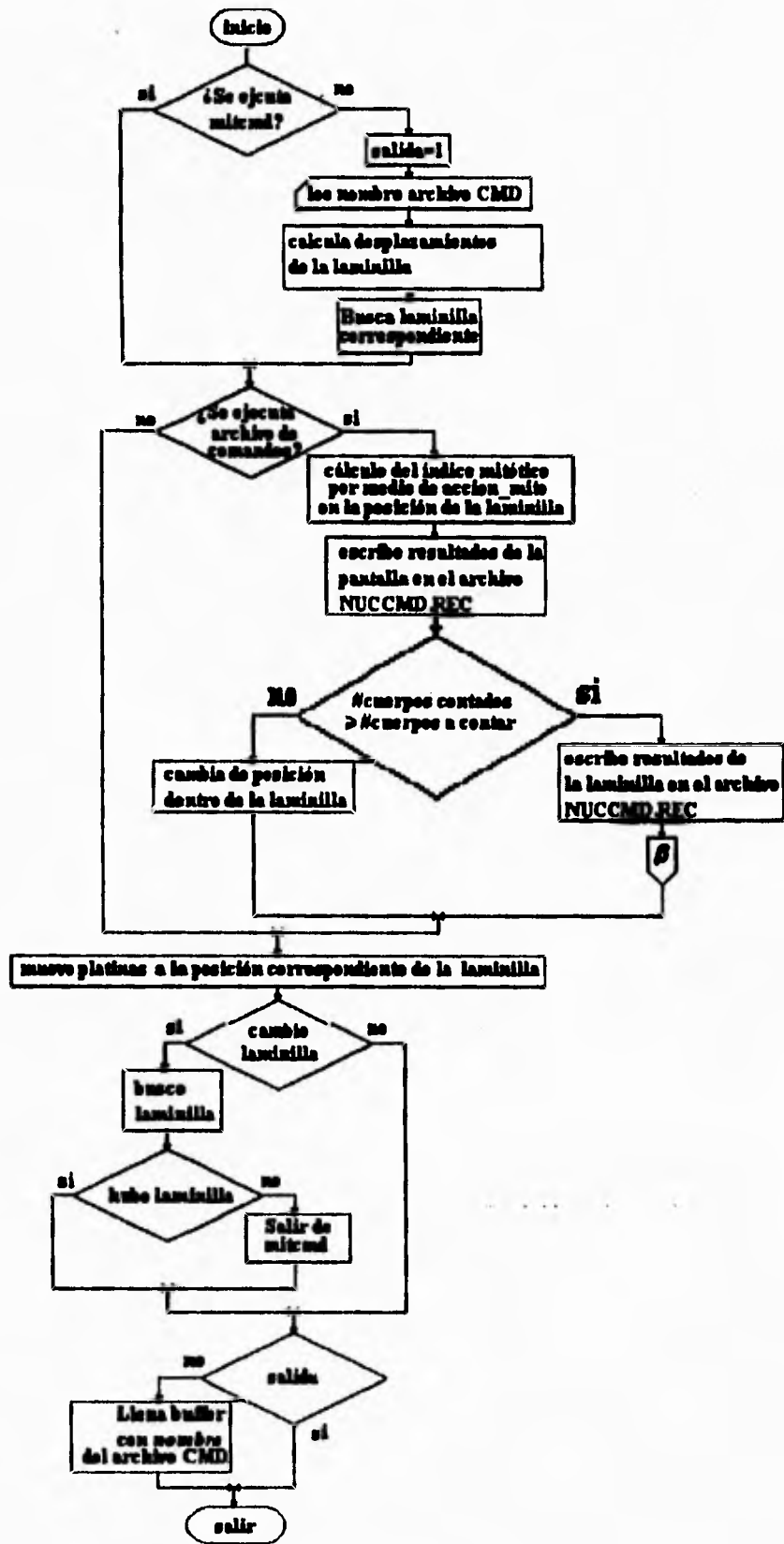


Fig. 4.41 Diagrama de flujo del funcionamiento de la rutina mitcmd.

De la misma forma que `leccmd` lee el archivo de comandos `mitcmd` realiza el mismo proceso, simplemente modifica el *buffer* físico. Las operaciones del análisis del índice mitótico se llevan a cabo de la misma forma en que se realizan en `mit` y siguen las especificaciones dadas por los usuarios, es decir: uso de controles y experimentos, número de cuerpos limitado, barrido en zigzag y enfoque. La coordinación de la operación entre el archivo de comandos y el análisis del índice mitótico por cada campo se realiza por medio de la variable `AC` (o `MITO.AC`).

Es importante mencionar que por el momento está vedado incluir dentro del archivo de comandos algunas rutinas del módulo de mitosis, pues podrían ocasionar resultados impredecibles, como recursión. También se debe mencionar que en `BIOCOM` hay forma de interrumpir la ejecución de un archivo de comandos por medio de oprimir `^Z`. Esto siempre se decodifica en la rutina `LECCOM`, que como ya se dijo es el decodificador de comandos. Para que las variables `BAN1`, `BAN2` y `AC` también regresarán a sus valores iniciales, hubo que modificar la rutina `LECCOM` para que al teclear `^Z`, estas tomaran el valor de cero.

IV.7. Rutina de verificación

IV.7.1. Archivo de Control

Para que sea posible revisar la operación del sistema durante la ejecución del análisis de campos, toda la información es escrita en un archivo; en el caso de `mit` se llama `NÚCLEOS.REC` y en el caso de `mitcmd` `NUCCMD.REC`. Este archivo va guardando la siguiente información:

ESTRUCTURA DEL ARCHIVO

	Campo1	Campo2	Campo3	Campo4	Campo5
Tipo	Caracter	Real	Real	Caracter	Caracter

La siguiente estructura se repite para cada portaobjetos instalado:

La siguiente estructura se repite para cada campo de análisis:

#portaobjetos	coordx	coordy	M	1
#portaobjetos	coordz	#pantalla	m	1
umbral de segmentación	#pantalla	0.0	u	1

categoria (e,s,a,c)	posx	posy	0	1
Letra p	cantidad de núcleos estimulados en la pantalla	cantidad de núcleos estimulados en la pantalla	p	1
Letra p	cantidad de cromosomas cerrados en la pantalla	cantidad de cromosomas abiertos en la pantalla	0	1
Letra p	cantidad de núcleos estimulados acumulados hasta esta pantalla	cantidad de núcleos estimulados acumulados hasta esta pantalla	P	1
Letra p	cantidad de cromosomas cerrados acumulados hasta esta pantalla	cantidad de cromosomas abiertos acumulados hasta esta pantalla	0	1

Al terminar el portaobjetos se escriben los resultados de la laminilla:

Letra r	cantidad de núcleos estimulados	cantidad de núcleos estimulados	r	1
Letra r	cantidad de cromosomas cerrados	cantidad de cromosomas abiertos	0	1

La filosofía del archivo es la siguiente: guardar la información de la posición de las platinas para cada campo muestreado, guardar la ubicación en la pantalla de cada cuerpo clasificado y guardar la cantidad de cada uno de los cuerpos reconocidos en cada campo analizado y en el total de la muestra.

Algunos campos del archivo demuestran el avance que fue sufriendo el desarrollo del sistema. Un ejemplo de esto, es la existencia de dos campos que guardan la cantidad de núcleos estimulados. Originalmente, como ya se indicó, se contaban dos categorías de núcleos: chicos y grandes.

También se puede observar que existe un campo que guarda el umbral de segmentación de cada pantalla. Al comienzo del desarrollo del sistema se pensó que en la rutina de verificación se usaran los mismos contornos que el sistema encontró al realizar el proceso del índice mitótico; esto se haría almacenando el nivel de gris con el cual se segmentó la imagen durante la ejecución del programa y así se obtendrían los mismos objetos. Sin embargo, posteriormente dejó de ser usado dicho nivel de gris debido al cambio que fue sufriendo la forma de segmentar; quizás el archivo debió modificarse para contener dos campos de umbral, uno para la primera segmentación y otro para la segmentación de las

mitosis dispersas, pero estos campos perdieron su importancia pues se decidió trabajar con símbolos en la pantalla que otorgan información sobre la posición y categoría de cada objeto clasificado.

```

A  AVANZAR UNA PANTALLA O EL BOTON DER DEL RATON
R  RETROCEDER UNA PANTALLA O EL BOTON IZQ DEL RATON
P  AVANZAR UN PORTAOBJETOS
C  CAMBIAR CATEGORIA
I  IR AL INICIO DEL PORTAOBJETOS N
F  MUESTRA LOS RESULTADOS DEL PORTAOBJETOS ACTUAL
M  MUEVE n PANTALLAS
T  MUESTRA LOS RESULTADOS TOTALES
B  BORRA LA PARTE GRAFICA
G  VUELVE A PONER LA PARTE GRAFICA
Esc PARA CANCELAR LA VERIFICACION
pos x:7525.400293 y:4640.964202      z:0.000000000 pantalla:1.000000000
Portaobjetos #1

      Resultados de pantalla:
Nucleos estimulados grandes:      0
Cromosomas abiertos:      0      Cromosomas cerrados:      2
      Acumulados:
Nucleos estimulados grandes:      0
Cromosomas abiertos:      0      Cromosomas cerrados:      2

```

Fig. 4.42 Pantalla de verificación

El campo número cuatro de la estructura de registros del archivo es el más importante para controlar la lectura del archivo durante la verificación de NÚCLEOS.REC y NUCCMD.REC, pues es por medio de este campo que se puede saber qué significado tienen los tres campos anteriores: posición de pantalla, posición del microscopio, y qué resultados hubo tanto en un campo de análisis como en cada laminilla.

El campo cinco del registro está implementado para poder hacer modificaciones al archivo. Es una bandera que avisa qué campo fue modificado, y debe ser borrado o actualizado.

IV.7.2. Rutina de verificación

Con respecto a la rutina de verificación cabe decir que ésta fue creada con el propósito de revisar el comportamiento del sistema al calcular el índice mitótico y, sobre todo, poder corregir los errores que el sistema cometió durante su proceso de análisis. Los errores pueden ser debidos a la lógica del programa o a la lógica del reconocimiento.

Como se mencionó al explicar la forma en que se reconocen los objetos, el filtro de cada categoría tiene un margen de error por el cual pueden entrar algunos elementos de otras categorías. Originalmente la rutina de reconocimiento fue planeada para poder corregir los errores de este tipo, sin embargo por la misma naturaleza de la rutina que funciona como una caja negra de los aeroplanos, permite llevar una bitacora de la operación del sistema lo que a *posteriori* posibilita la corrección de errores en el funcionamiento del programa.

La rutina básica que se encarga de la revisión de la bitacora es *verify*. Esta rutina se puede esquematizar como sigue: Primero se selecciona el archivo de verificación que se desea revisar y luego se seleccionan una serie de operaciones que permiten verificar de forma interactiva el accionar de la rutina para el cálculo del índice mitótico.

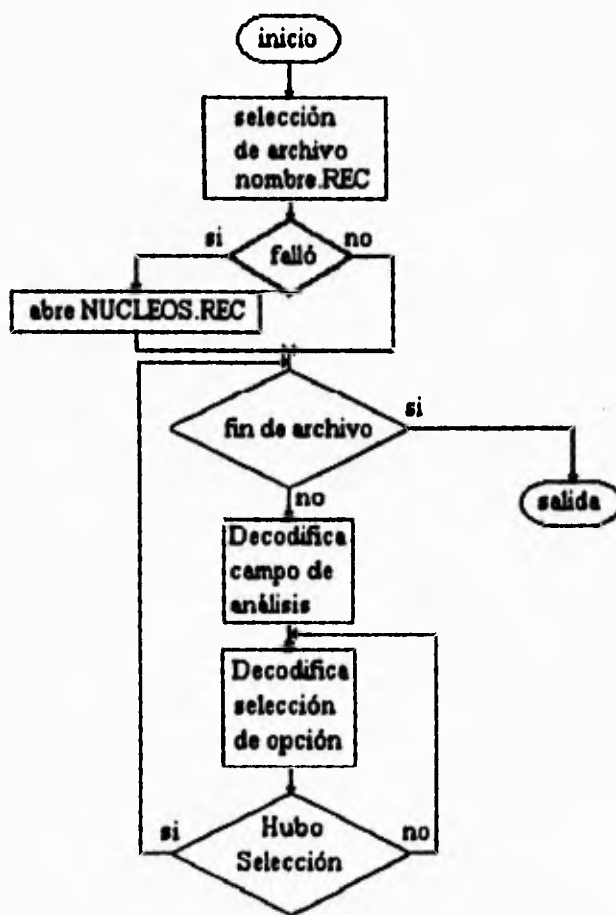


Fig. 4.43 Diagram de flujo del funcionamiento de *verify*

La sección que decodifica el campo de análisis se refiere a la parte del archivo:

#portaobjetos	coordx	coordy	M	1
#portaobjetos	coordz	#pantalla	m	1
umbral de segmentación	#pantalla	0.0	u	1
categoría (e,s,a,c)	posx	posy	0	1
Letra p	cantidad de núcleos estimulados en la pantalla	cantidad de núcleos estimulados en la pantalla	p	1
Letra p	cantidad de cromosomas cerrados en la pantalla	cantidad de cromosomas abiertos en la pantalla	0	1
Letra p	cantidad de núcleos estimulados acumulados hasta esta pantalla	cantidad de núcleos estimulados acumulados hasta esta pantalla	P	1
Letra p	cantidad de cromosomas cerrados acumulados hasta esta pantalla	cantidad de cromosomas abiertos acumulados hasta esta pantalla	0	1

Esta sección imprime en pantalla de texto los resultados, el número y las coordenadas del campo de análisis correspondiente.

La parte que selecciona opciones se refiere a que en la rutina verify existen una serie de opciones (figura 4.42) para recorrer el archivo que se pueden enumerar como sigue:

- Avance por campos continuos
- Retroceso al campo anterior
- Avance a campos donde se halló cierta categoría
- Cambio de portaobjetos
- Avance de cierto número de campos de análisis
- Ir al inicio del portaobjetos n
- Muestra los resultados de la laminilla en revisión
- Borra la parte gráfica del sistema
- Regresa la parte gráfica
- Cancela el proceso de revisión
- Modifica algún objeto hallado (eliminación o cambio)
- Agrega algún objeto faltante

Estas rutinas pueden resumirse en la siguiente forma:

1. Leer posición actual del archivo.
2. Escoger llave de búsqueda u operación.
3. Si fue operación no regresa llave de búsqueda modificada de lo contrario regresa la llave modificada.
4. Salir.

Las llaves de búsqueda varían según la opción que se desea. Sin embargo hay ciertas excepciones. Por ejemplo, cuando se quiere avanzar a campos con cierta categoría se usa como llave el primer campo, que contiene la letra que representa a dicha categoría, y el campo cinco que debe estar vacío; ambos después de los registros de movimiento para las platinas.

Casos especiales son: el cambio, inserción y eliminación de objetos. La inserción todavía no está finalizada pero las otras dos modifican los elementos por medio de la llave campo cinco. Si el campo cinco con cierta categoría está apagada, entonces ese objeto será eliminado del archivo.

IV.7.3. Auxilio Extra de Verify.

Esta rutina, a pesar de estar planeada para realizar la corrección de los errores de reconocimiento, permitió hallar errores en la programación y errores de implementación física.

Además ayudó a descubrir algunos problemas, uno de ellos fue la saturación de contornos y/o puntos dentro de una imagen. Pues, como ya se mencionó, el archivo REC guarda el nivel de gris que sirvió como umbral para segmentar una imagen. Con esta información y con el conocimiento de la forma de realizar la segmentación se pudo ver que el proceso de segmentación no era el óptimo y debía reformarse.

También mostró la necesidad de que era necesario aumentar el contraste en algunas imágenes debido a que en ellas la diferencia entre el fondo y los objetos era muy tenue, provocando que el método de OTSU encontrara un nivel de gris que detectaba contornos que no encerraban a los objetos de interés. De aquí surgió la idea de usar las LUT's para aumentar el contraste.

Otro problema que ayudó a encontrar fue el problema de enfoque. En este caso el sistema guardaba posiciones en el eje Z que no concordaban con la imagen en foco. La razón podría provenir de dos fuentes: una la interfaz con el controlador MultiControl 2000 estaba funcionando mal, y dos, que el microscopio tenía mal una unión con el motor de pasos que controla el eje Z. Después de algunas pruebas de movimiento se descubrió que la causa del problema era la unión física del microscopio y el motor de pasos. Esta unión era por medio de una goma de hule que hacía fricción con el pivote de enfoque; este tipo de unión provocaba que se perdiera contacto cuando se llevaba al cabo un movimiento brusco del motor de pasos, así que había que mejorar la unión. Ahora la unión (figura 4.44) está hecha por presión de manera provisional mientras se monta el equipo en su laboratorio final.

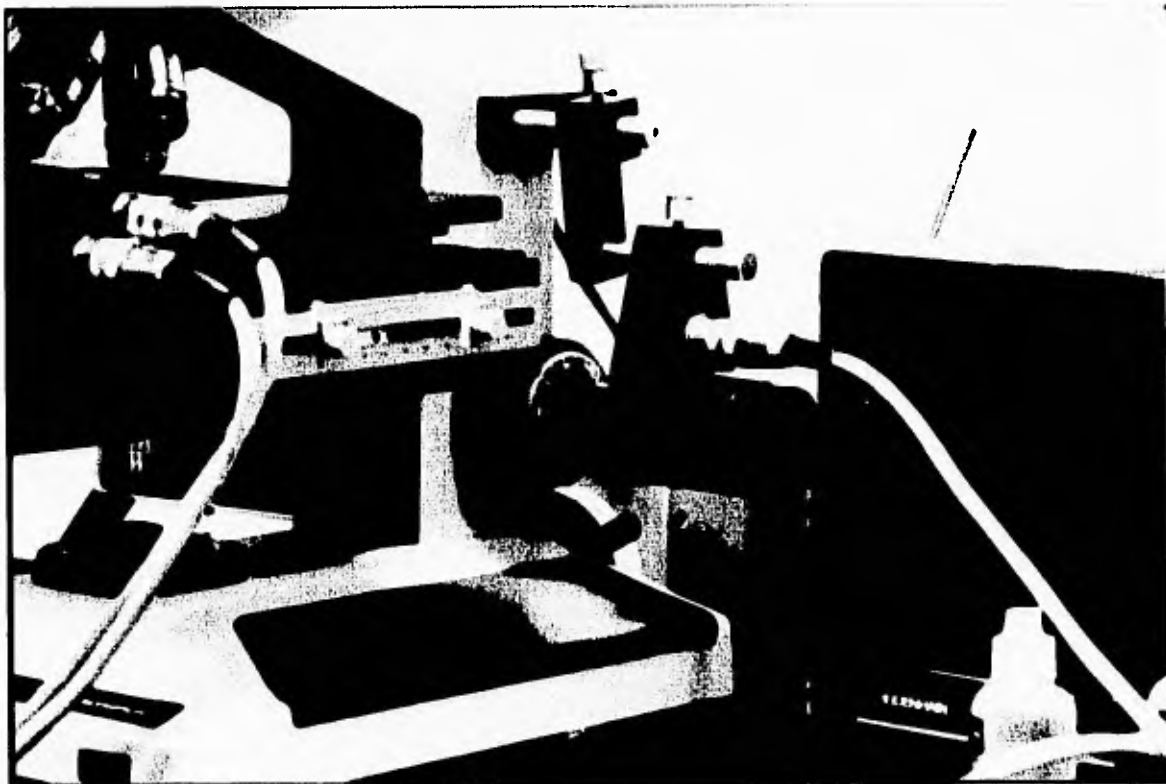


Fig. 4.44 Diagrama de uniones del eje z del microscopio

Obviamente la rutina ayudó a refinar los filtros de reconocimiento hasta su estado actual.

CAPITULO V

PRUEBAS Y AJUSTES DEL PROTOTIPO

V.1. Interfaz con BIOCOM

Comencemos por la parte de la interfaz con BIOCOM. En este caso el sistema no ha tenido ningún problema pues las rutinas han sido bien alojadas y no se tiene conflicto con el uso de la memoria. Sin embargo las rutinas para el cálculo del índice mitótico son muy grandes, en código objeto, así que se debe ser cuidadoso sobre el tamaño de la pila (*stack*) y del tamaño de cada modulo de *overlay*.

V.2. Interfaz con el MultiControl 2000

La forma de barrido de las platinas no ha experimentado problemas y, por el contrario, las opciones sobre nuevas definiciones de barrido han agregado flexibilidad al análisis de los campos.

En cambio, se han tenido problemas con la interfaz que se hace con el controlador **MultiControl 2000**, que actualmente está funcionando a una velocidad de comunicación entre computadora y controlador, de 1200 bauds. Aunque teóricamente el sistema permite el trabajo con 9600 bauds como máximo. Durante la implementación jamás se pudo hacer que ambas máquinas, controlador y computadora, pudieran establecer comunicación a la velocidad de 9600 bauds que supuestamente es la velocidad máxima que soporta el controlador, así que se estableció de 4800 bauds.

A esta velocidad se tienen los mismos problemas que actualmente se tienen a 1200 bauds y el rendimiento del sistema es casi el mismo, pues las operaciones que más toman tiempo, hasta ahora, son las del procesamiento de imágenes tanto carga y descarga de imágenes como operaciones morfológicas y afinamientos). Es por ello que no sea ha cambiado la velocidad a 4800 bauds.

Existe otro problema con el controlador y es que después de cierto número de operaciones, la comunicación se interrumpe por culpa del controlador pues deja de reconocer los comandos mandados desde la computadora. Al parecer sufre una saturación en el buffer que recibe los comandos; hasta ahora no se ha encontrado nada escrito sobre este problema en la documentación del **MultiControl 2000**.

Junto a este problema sucede que súbitamente el controlador deja de recibir comandos y comienza a moverse con un patrón repetitivo que no fue mandado con

anterioridad. Esto lleva a pensar que existe una saturación del controlador. Este problema debe ser verificado en el siguiente desarrollo del sistema.

V.3. Archivos de Control

De la explicación del funcionamiento de las rutinas asociadas al cálculo del índice mitótico se deduce que no pueden crear archivos distintos a NÚCLEOS.REC y NUCCMD.REC. Este problema será solucionado posteriormente pues no representa un obstáculo para la operación del sistema.

V.4. Rutinas de lectura de Archivos de Comandos

Hay que decir que aun hace falta realizar más pruebas sobre el buen funcionamiento de las rutinas que combinan la ejecución de un archivo de comandos y las rutinas de movimiento y análisis de cada campo en las laminillas (leecmd y mitcmd) pues las pruebas hasta ahora han sido limitadas. La limitación en el número de pruebas se debe, básicamente, a que la evaluación del sistema se basó en la prueba de la rutina mit pues es la que abarca la mayor parte del interés del sistema.

V.5. Rutinas de Verificación de Ejecución del Índice Mitótico

Cuando se realiza la verificación del reconocimiento que hizo el sistema se puede ver que existe la opción de agregar un objeto faltante. Hasta ahora esta parte del sistema aún no ha sido implementada debido a que se considera más importante hacer que el sistema cometa la menor cantidad de errores. La causa de que las inserciones no se han llevado a cabo es porque estas agregan un problema extra en el manejo de archivos en el lenguaje C. Para resolver este problema se han manejado listas ligadas sobre la memoria primaria de la computadora y puede que esta memoria sea limitada, recordar el límite de los 640Kbytes en las PC y el manejo de listas implica que se pueda borrar algún objeto que anteriormente fue insertado.

V.6. Proceso de Reconocimiento

Hablando sobre el estudio que se hace sobre cada campo de análisis se puede decir que los procesos que se ven involucrados: proceso de enfoque sobre un campo, análisis del mismo y vaciado de resultados, se están llevando a cabo por el sistema en un tiempo máximo de treinta segundos, dependiendo de la calidad de la imagen. Este tiempo de proceso es bastante bueno ya que originalmente se había puesto como meta el realizar este conjunto de procesos en un máximo de treinta y cinco segundos para que el sistema tuviera un

desempeño aceptable bajo las restricciones de tiempo que los biólogos pusieron¹. Si por cada toma se halla un promedio de diez objetos de interés entonces cada laminilla, muestreando dos mil objetos, es analizada en más o menos una hora y cuarenta minutos, lo que todavía es un tiempo alto pues la meta es lograr que el sistema tome una hora en dicho análisis, lo que implica alrededor de veinte segundos por toma.

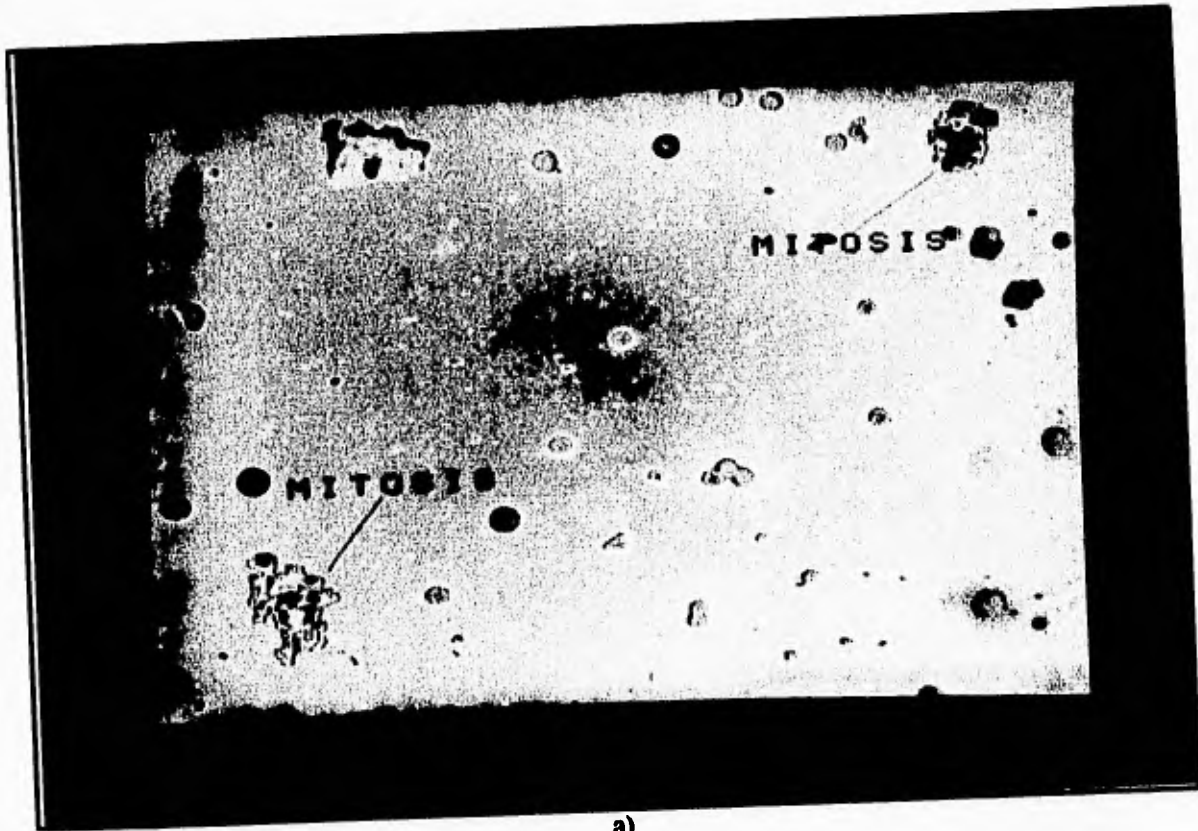
V.7. Resultados del Reconocimiento

Los resultados del reconocimiento de los objetos fueron evaluados siguiendo el procedimiento siguiente: el equipo era dejado funcionar con laminillas que posteriormente serían revisadas usando la rutina de verificación. La operación de verificación ocasionalmente era realizada en compañía de biólogos, este requisito no fue necesario todo el tiempo ya que las operaciones de entrenamiento permitieron obtener experiencia sobre los cuerpos que deben ser clasificados y cuales no.

Durante la fase verificación se examinaron alrededor de 10 laminillas diferentes y en cada una de ellas unos dos mil núcleos, lo que resulta en más o menos 20, 000 cuerpos.

Los resultados que ofrece el sistema han sido considerados buenos (figura 5.1), debido a que se tiene un alto porcentaje en el reconocimiento positivo de cada una de las categorías: 91% en el caso de los núcleos estimulados, 80% en el caso de las mitosis abiertas y 70% en el caso de las mitosis cerradas cuando se realiza un entrenamiento sobre laminillas que no difieren mucho de las que serán analizadas.

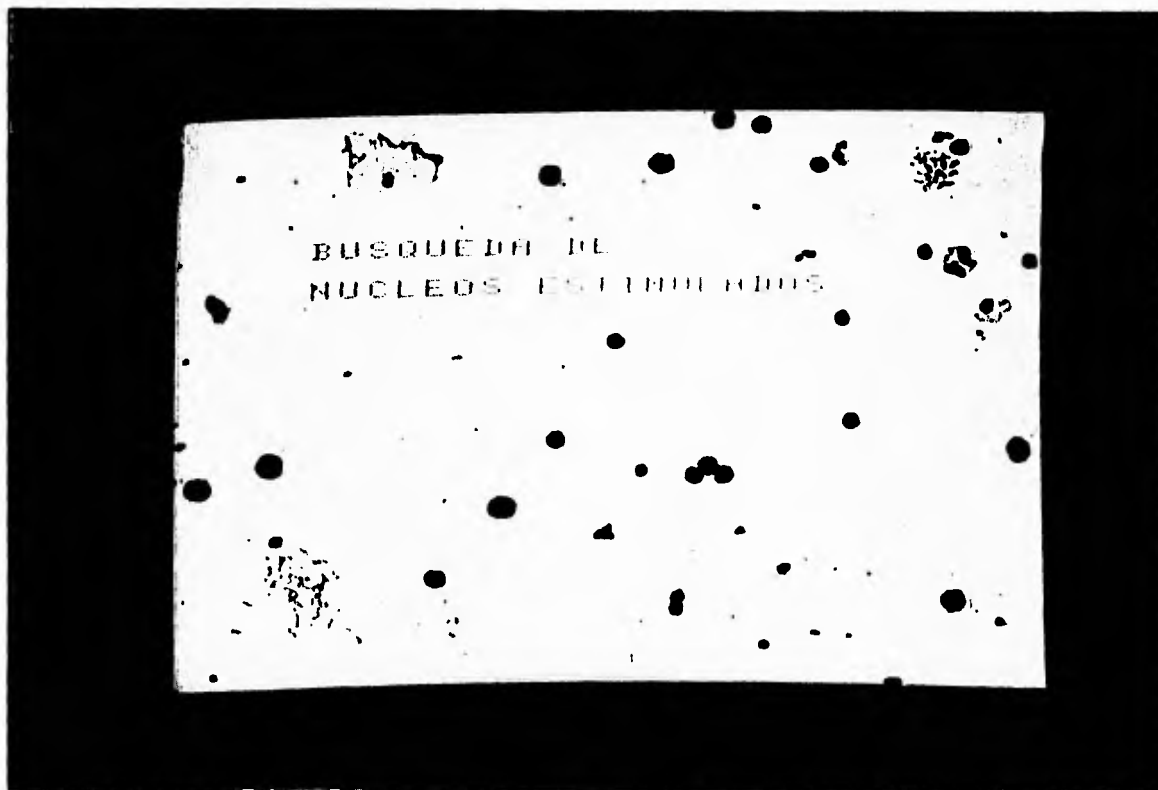
¹recordar que el tiempo que toma el análisis manual de una laminilla está entre veinte y cuarenta minutos, dependiendo de la experiencia que tenga el observador.



a)
Fig. 5.1. Proceso de reconocimiento. a) Imagen original. b) Reconocimiento de mitosis compactas.
c) Reconocimiento de núcleos estimulados. d) Reconocimiento de mitosis dispersas.
e) Resultado final.

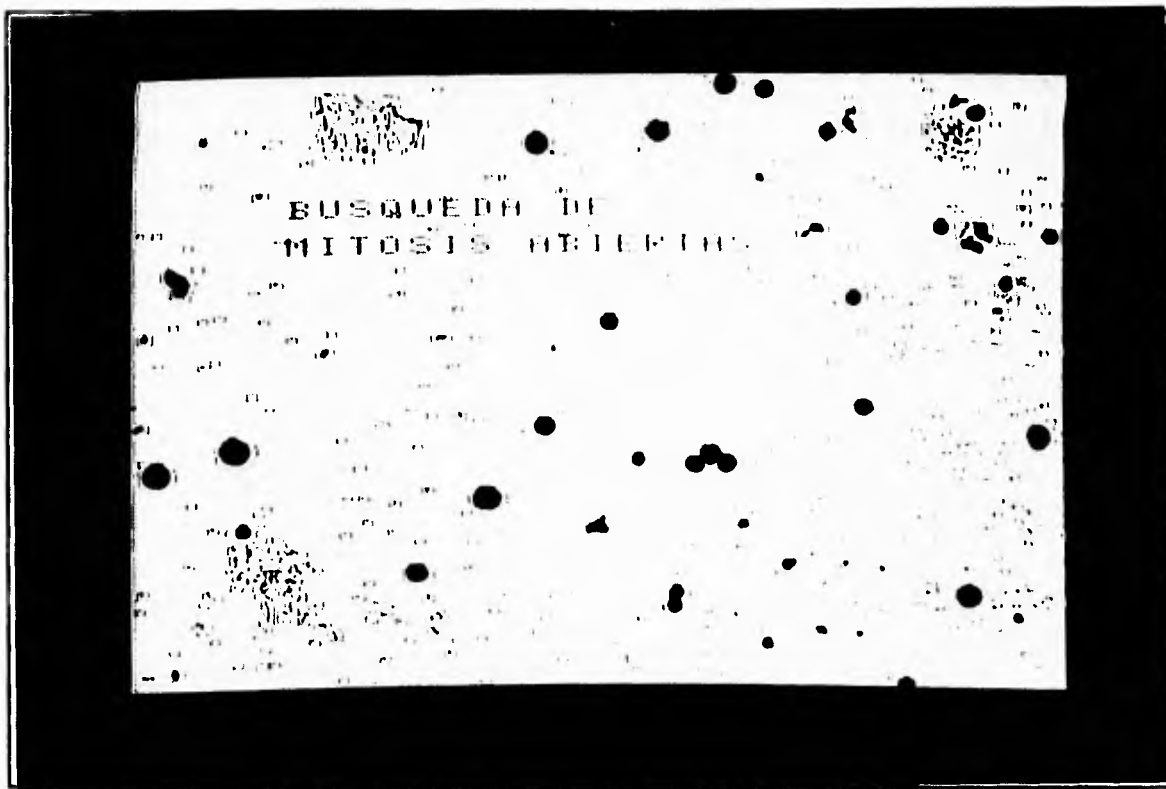


b)

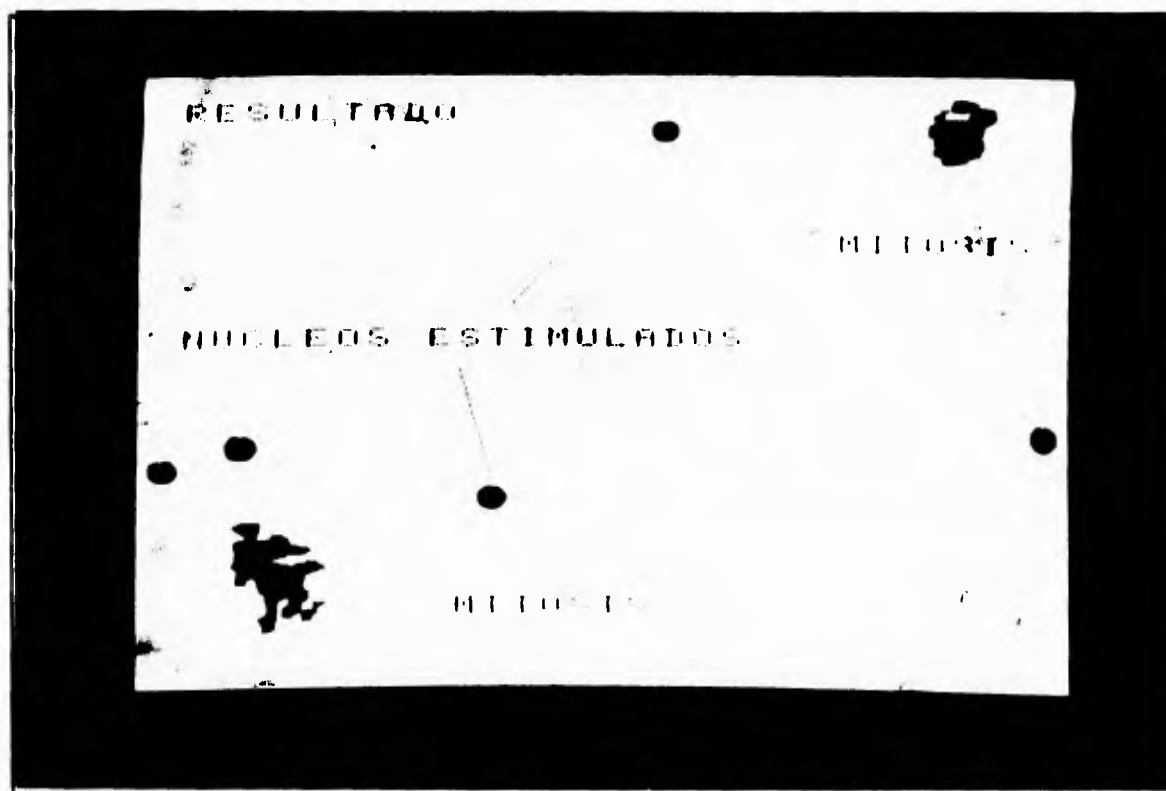


c)

Fig.5.1. Proceso de reconocimiento. a)Imagen original. b) Reconocimiento de mitosis compactas. c) Reconocimiento de núcleos estimulados. d) Reconocimiento de mitosis dispersas. e) Resultado final.



d)



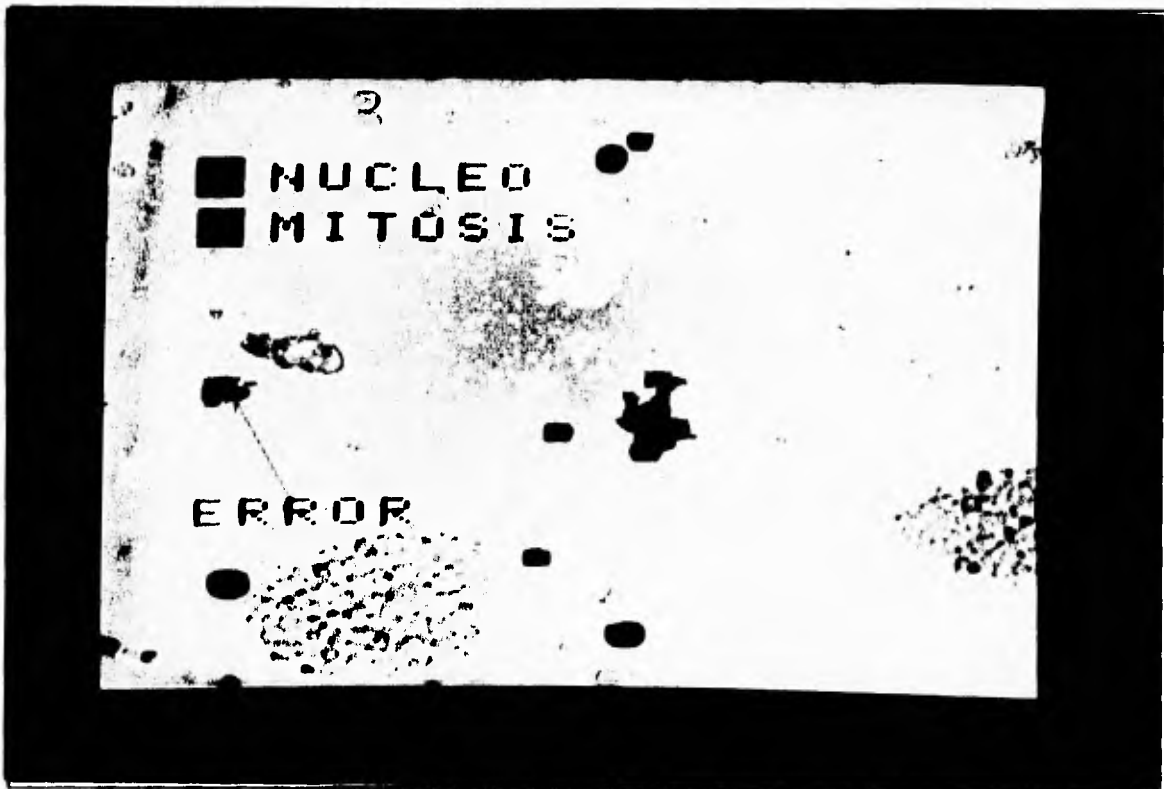
e)

Fig.5.1. Proceso de reconocimiento. a)Imagen original. b) Reconocimiento de mitosis compactas. c) Reconocimiento de núcleos estimulados. d) Reconocimiento de mitosis dispersas. e) Resultado final.

V.8. Errores de Reconocimiento

Como se deduce de las cifras anteriores, el sistema tiene un mayor porcentaje de error en el reconocimiento de las mitosis cerradas, esto se debe a que en éstas los parámetros implementados no han sido lo suficientemente definitivos como para separar las mitosis cerradas de los demás objetos que se hallan en la pantalla de análisis (figura 5.2).

Por medio de la experimentación se ha observado que la medición de la textura usando los cambios de los niveles de gris al interior del contorno puede no ser la falla al clasificar los objetos, sino probablemente sea el cómo se están realizando las mediciones de textura. Esta deducción parte del hecho que los artefactos que llegan a entrar dentro del conteo de mitosis son diametralmente distintas a las mitosis compactas, por lo menos para el ojo humano. Se cree que al tomar como nivel de referencia al obtenido por el método de Otsu lo único que logra es mediatizar los cambios haciéndolos muy parecidos para todos los cuerpos.



a)

Fig. 5.2. Errores en el reconocimiento. a) ejemplo 1. b) ejemplo 2.



b)

Fig. 5.2. Errores en el reconocimiento. a) ejemplo 1. b) ejemplo 2.

En el caso de las mitosis abiertas se logró encontrar un parámetro extra que permite la separación de las mitosis abiertas de la mayoría de los otros objetos. Sin embargo, se tienen todavía problemas de inclusión de artefactos que pudiera evitarse si es que fuera posible medir mejor la textura de la mitosis; es decir, si se mide mejor la orografía del contorno.

Sobre cómo reconocer los distintos cuerpos que se hallan en una muestra no existe nada escrito. Es decir no hay un método que nos permita saber cuáles son las características de un núcleo o de una mitosis, así que la mayor parte del tiempo del desarrollo del sistema se basó en encontrar la mejor forma de realizar la clasificación de dichos cuerpos.

Como se menciona en el capítulo anterior, se tuvieron que hallar los parámetros que caracterizaban a cada uno de los objetos. Cabe señalar que la falla en el reconocimiento de algún núcleo no es muy significativo, pues casi todos los objetos de las muestras son núcleos o artefactos así que la exclusión de uno de ellos puede no modificar significativamente los resultados.

En contraparte se encuentran las mitosis. Para que el ser humano reconozca a una mitosis en la laminilla se realiza un proceso muy complicado: es el reconocimiento de los componentes individuales de una mitosis y por medio de la textura del objeto. En ocasiones, para reconocer a una mitosis es necesario jugar con el enfoque del microscopio para que con

esta acción se tenga un mayor contraste y mayor profundidad de campo, o sea mayor información.

Sobre la textura existe mucho escrito: desviación en el nivel de gris en una zona, orientación de los píxeles, etcétera. Sin embargo, las mitosis presentan una textura que no posee un patrón definido y que se distinguen, manualmente, gracias a que es posible reconocer a los cromosomas o por lo menos parte de ellos, y porque presentan además una textura distinta a los artefactos.

El intento de hallar la textura básica de una mitosis fue la parte que llevó mayor tiempo de investigación. La decisión sobre cómo cuantificar la textura dentro de las mitosis fue ver, básicamente, la orografía que presentaban dichos cuerpos (es por ello que se decidió el contar los cambios en el nivel de gris).

Este parámetro ha funcionado, pero hay que decir que, como depende de un nivel de gris que fue hallado para dividir la distribución de niveles de grises en dos partes similares, se tiene el problema de que los cambios que se miden al interior de los objetos serán muy parecidos para muchos objetos, lo que ocasiona que muchos artefactos sean contados como mitosis. Sin embargo, la idea parece ser buena siempre que se halle la forma de medir de forma eficiente los cambios y la forma en que están distribuidos los niveles de grises al interior de los contornos.

Para evitar este problema se ha decidido modificar el nivel de referencia y realizar una búsqueda por niveles; es decir tomar una serie de líneas divisorias a cierto nivel de gris y cuantificar la cantidad de píxeles que se encuentran dentro de cada sección y la forma en que cambian de sección dichos píxeles, como se puede ver en la siguiente figura.

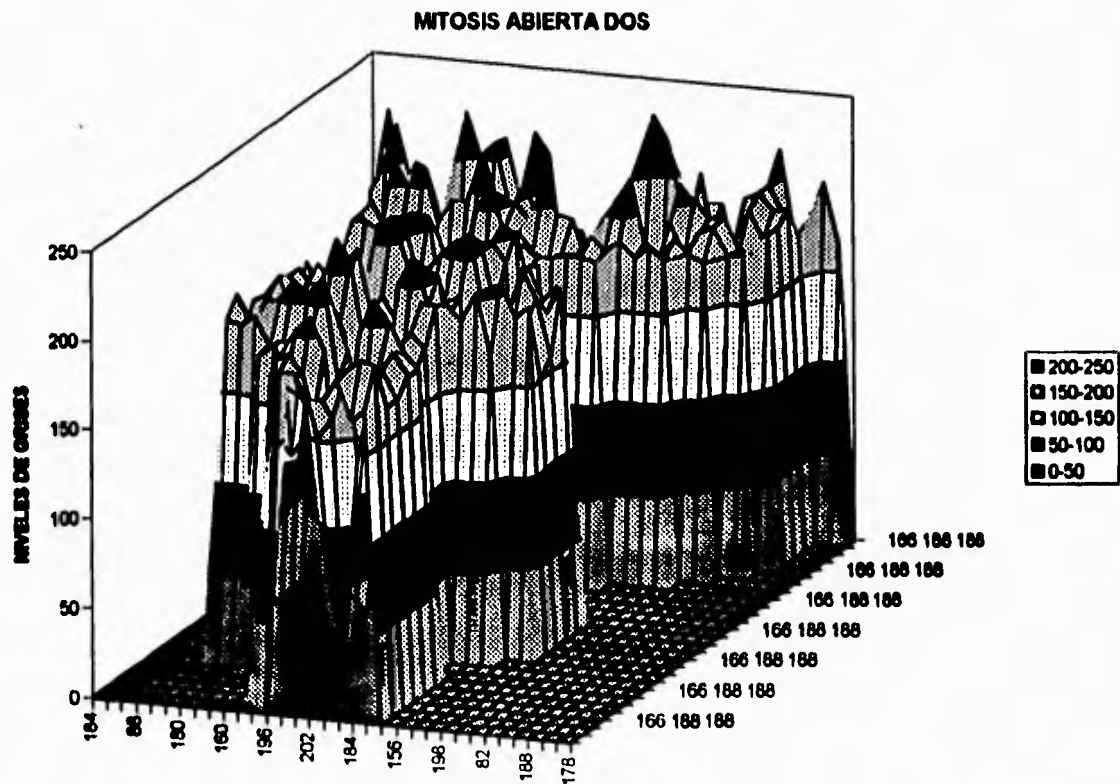
esta acción se tenga un mayor contraste y mayor profundidad de campo, o sea mayor información.

Sobre la textura existe mucho escrito: desviación en el nivel de gris en una zona, orientación de los pixeles, etcétera. Sin embargo, las mitosis presentan una textura que no posee un patrón definido y que se distinguen, manualmente, gracias a que es posible reconocer a los cromosomas o por lo menos parte de ellos, y porque presentan además una textura distinta a los artefactos.

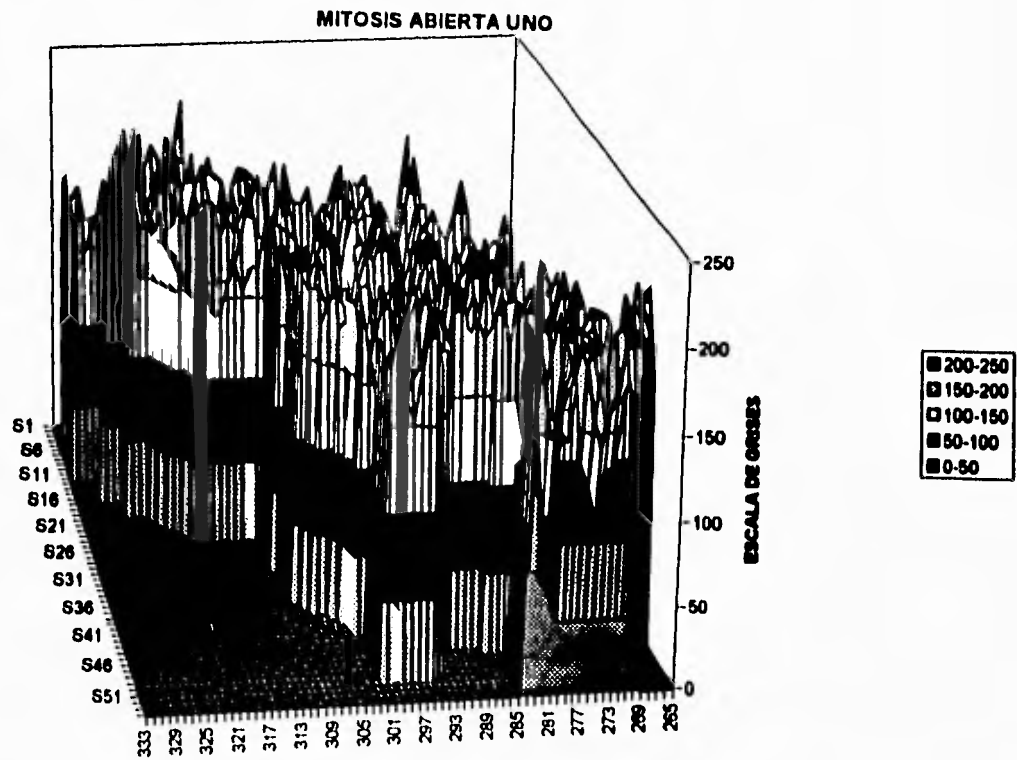
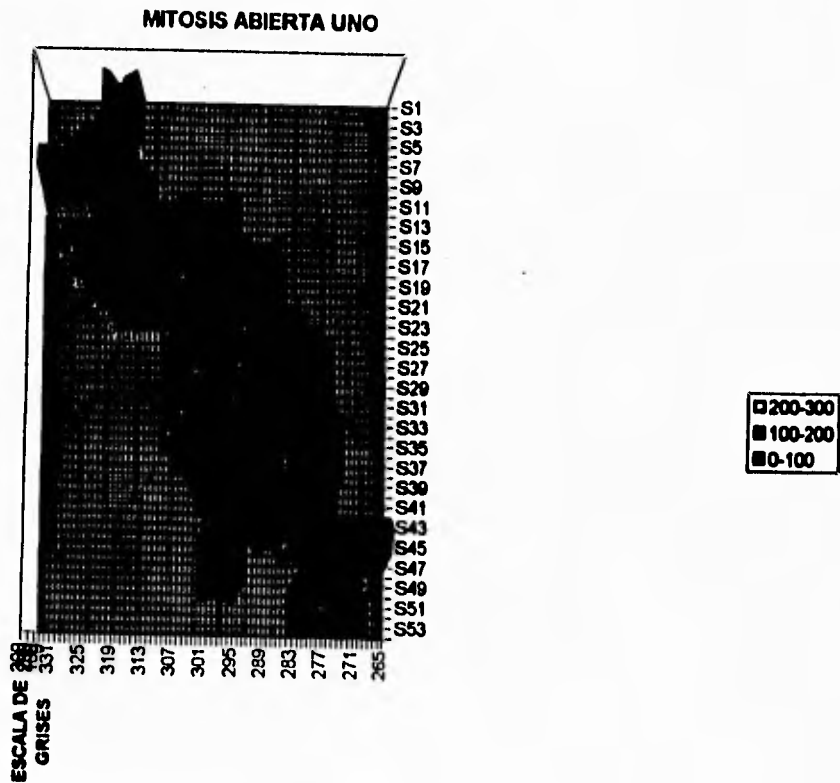
El intento de hallar la textura básica de una mitosis fue la parte que llevó mayor tiempo de investigación. La decisión sobre cómo cuantificar la textura dentro de las mitosis fue ver, básicamente, la orografía que presentaban dichos cuerpos (es por ello que se decidió el contar los cambios en el nivel de gris).

Este parámetro ha funcionado, pero hay que decir que, como depende de un nivel de gris que fue hallado para dividir la distribución de niveles de grises en dos partes similares, se tiene el problema de que los cambios que se miden al interior de los objetos serán muy parecidos para muchos objetos, lo que ocasiona que muchos artefactos sean contados como mitosis. Sin embargo, la idea parece ser buena siempre que se halle la forma de medir de forma eficiente los cambios y la forma en que están distribuidos los niveles de grises al interior de los contornos.

Para evitar este problema se ha decidido modificar el nivel de referencia y realizar una búsqueda por niveles; es decir tomar una serie de líneas divisorias a cierto nivel de gris y cuantificar la cantidad de pixeles que se encuentran dentro de cada sección y la forma en que cambian de sección dichos pixeles, como se puede ver en la siguiente figura.



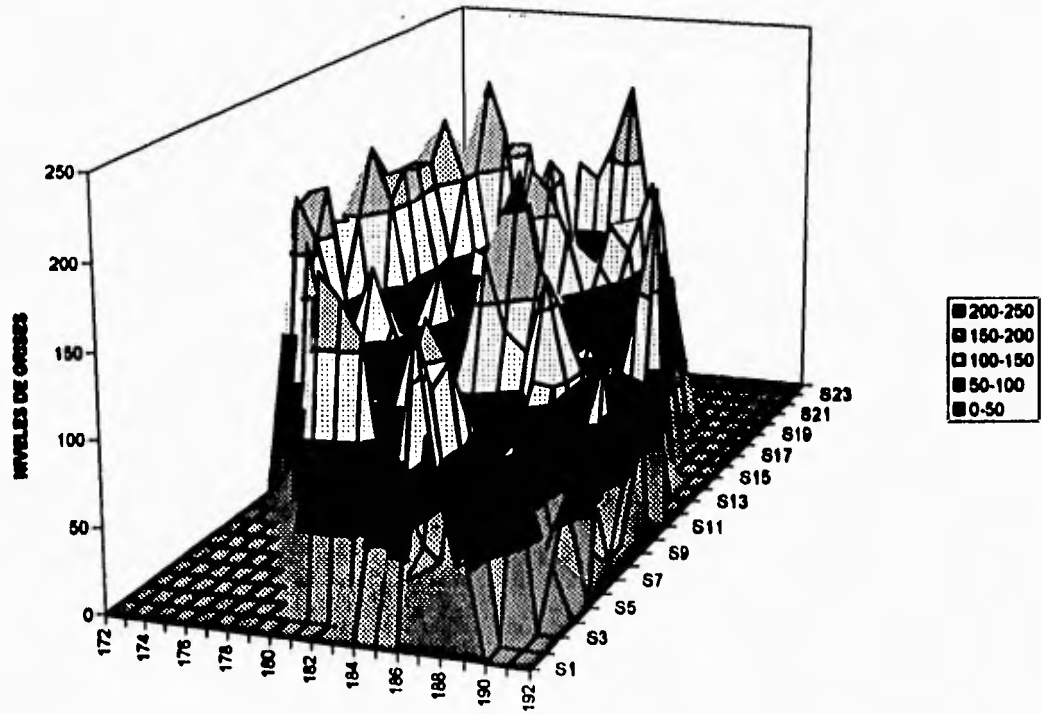
Haciendo algunas reconstrucciones tridimensionales del comportamiento de los pixeles dentro de un cuerpo (ver figura 5.4) se halló que las mitosis abiertas guardan un patrón en su distribución de niveles de grises; sin embargo, la investigación se encuentra en la etapa de realizar mayores reconstrucciones en 3D de los artefactos y de las mitosis para poder ver si la hipótesis es correcta o no.



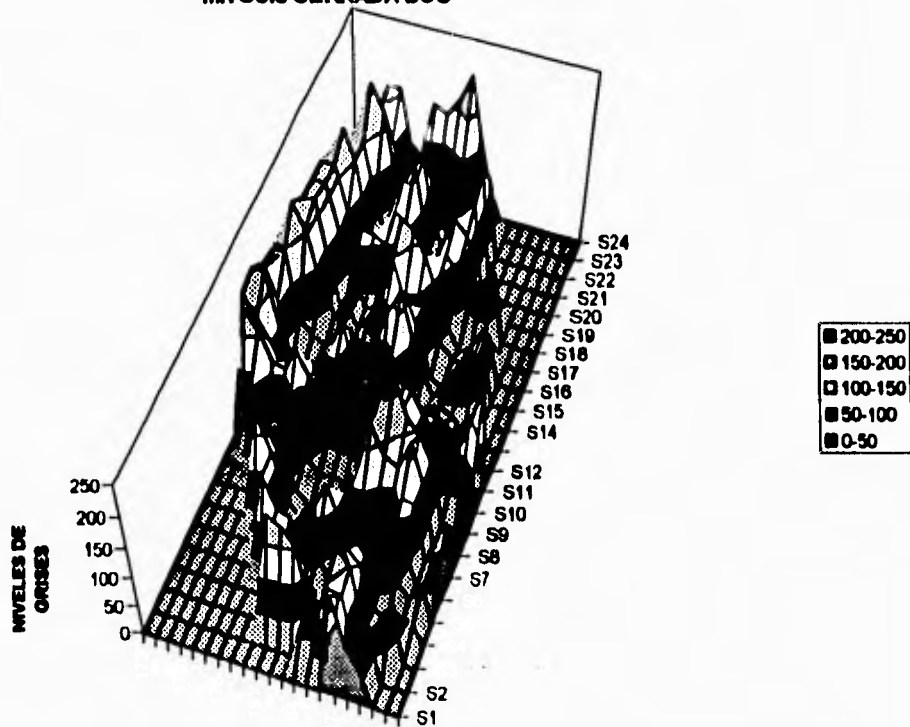
a)

Fig. 5.4. Reconstrucciones en 3D. a) Mitosis dispersas. b) Mitosis compactas. c) Artefactos, fase mitosis compactas. d) Artefactos, fase mitosis dispersas

MITOSIS CERRADA DOS

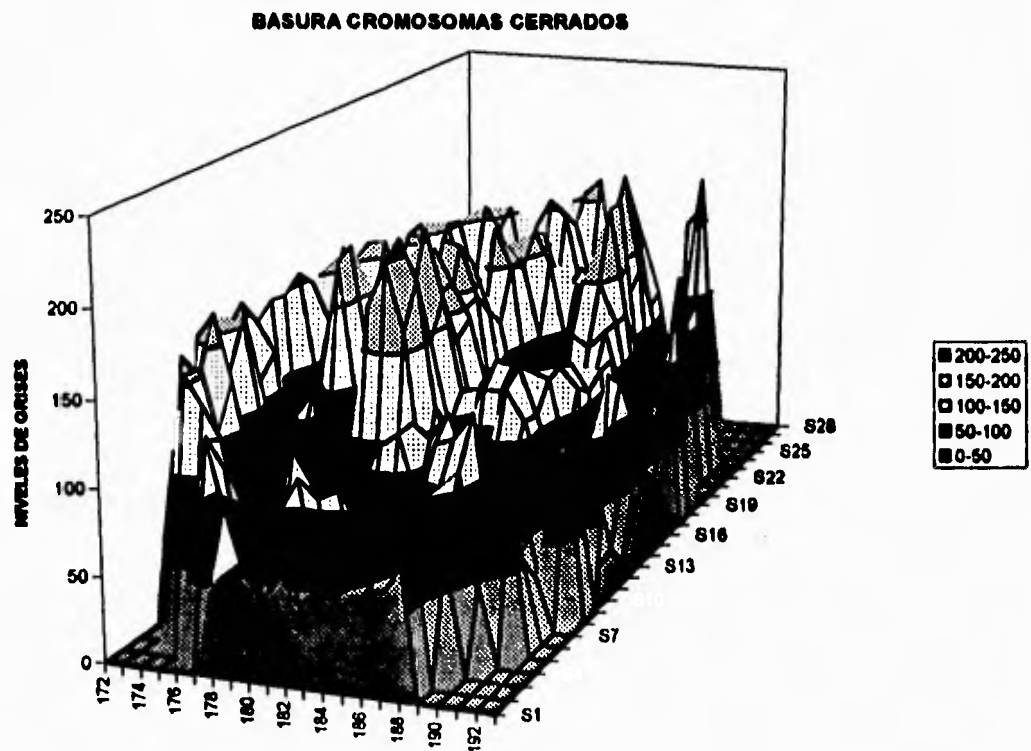


MITOSIS CERRADA DOS

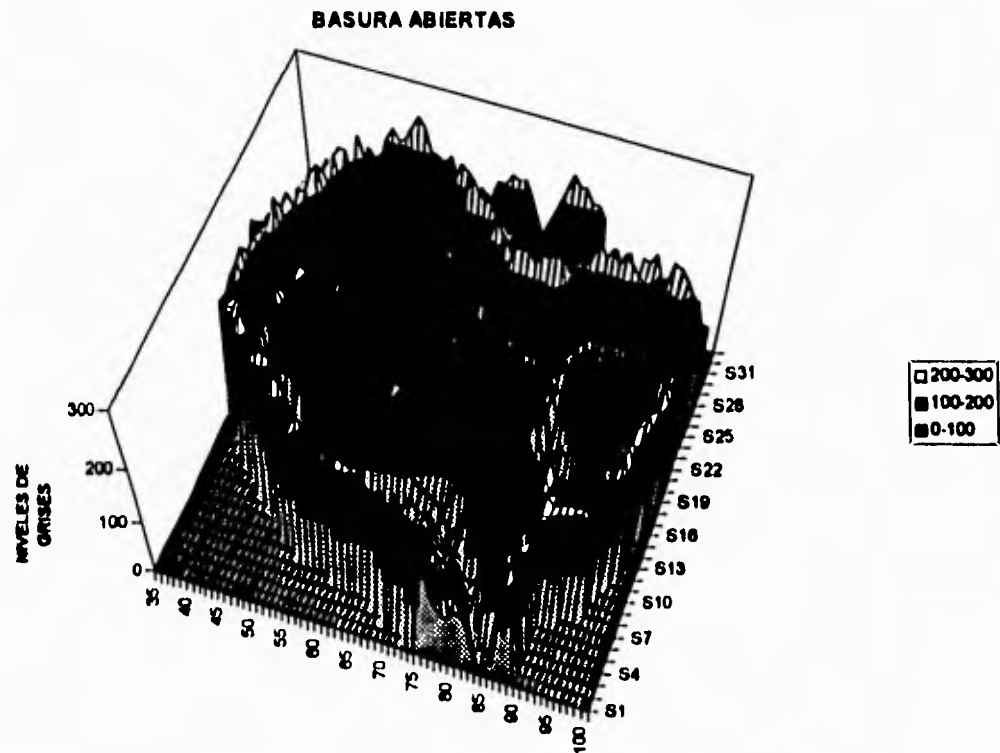


b)

Fig. 5.4. Reconstrucciones en 3D. a) Mitosis dispersas. b) Mitosis compactas. c) Artefactos, fase mitosis compactas. d) Artefactos, fase mitosis dispersas



c)



d)

Fig. 5.4. Reconstrucciones en 3D. a) Mitosis dispersas. b) Mitosis compactas. c) Artefactos, fase mitosis compactas. d) Artefactos, fase mitosis dispersas

V.8.1. Filtros Discriminantes

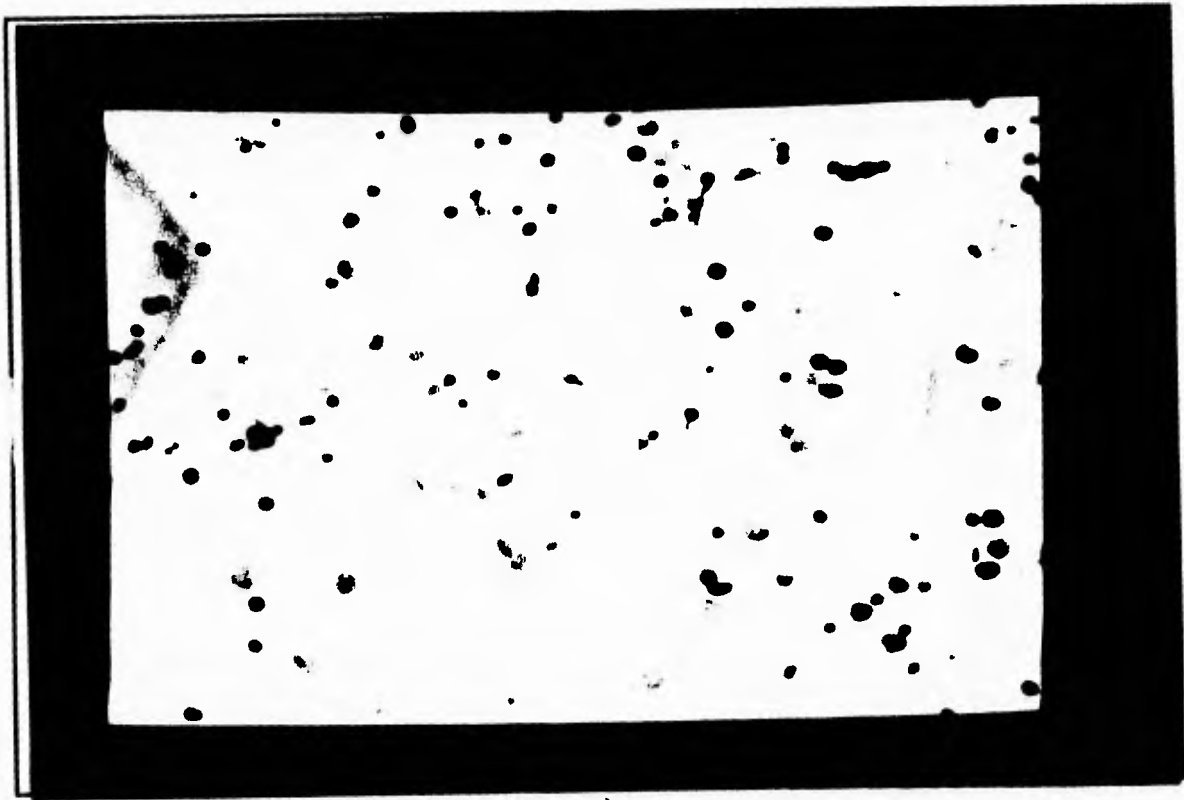
Sobre el problema del reconocimiento existe otra dificultad consistente en hallar la forma óptima de los filtros; es decir, el número de desviaciones estándar que debe ser usado para cada parámetro que conforma a cada filtro (ver figura 4.30). Esta medida ha sido realizada a nivel experimental a partir de los resultados obtenidos de las distintas revisiones del comportamiento del sistema por medio de la rutina *verify*. Sin embargo, se deben llevar a cabo mayores pruebas y análisis para lograr el nivel óptimo de cada filtro.

Para estos análisis se tienen planeadas otras etapas, que básicamente pueden dividirse en dos: una etapa de análisis que revise todas las posibilidades de inclusión de cada objeto, jugando con todas las combinaciones posibles de desviaciones estándar para cada objeto, y la segunda es el uso de redes neuronales para el reconocimiento de los cuerpos.

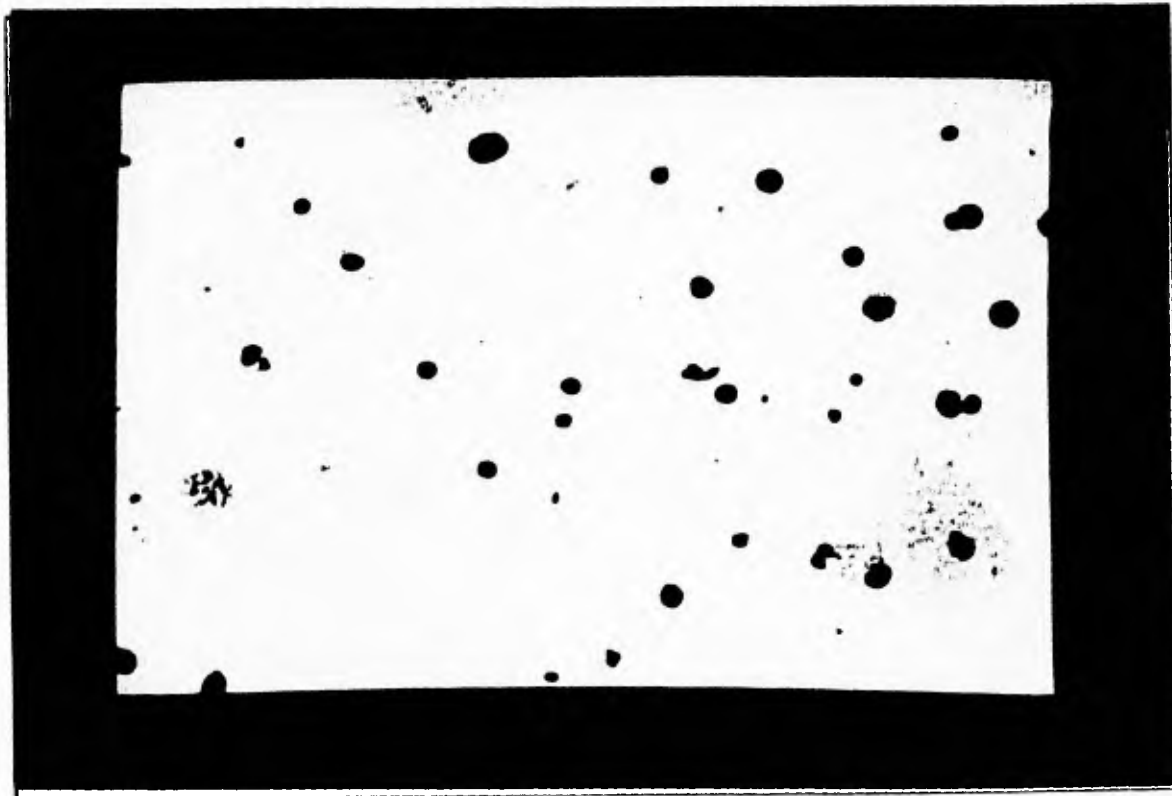
Nuevamente se debe evaluar si el subsistema de redes neuronales es lo suficientemente poderoso, capaz y rápido que ajuste con las especificaciones del sistema. Cabe mencionar que las redes neuronales funcionan muy bien para un entrenamiento pero las muestras que se manejan para el cálculo del índice mitótico tienen un espectro muy amplio de situaciones: sucias, claras, oscuras, muy pobladas, con mucha concentración del cultivo, etcetera, como ejemplo basta ver lo distintos que pueden ser la laminillas de un solo lote (ver figura 5.5). Es por ello que el sistema MIT 2000 tiene la opción de poder entrenar el sistema de distintas maneras y guardar dichos entrenamientos en archivos.

La otra vertiente requiere de realizar una buena planeación del experimento de análisis: tomar una muestra considerable de artefactos y objetos de interés en cada etapa de análisis, decidir el número de desviaciones estándar tope que serán probadas.

Dentro de los factores que han influido en las fallas en el reconocimiento de los objetos, se ha encontrado que la variabilidad de las muestras puede llegar a ser enorme, lo que provoca que los límites de desviaciones estándar usados para una muestra sean inútiles para otra. La diferencia en las preparaciones se debe muchas veces a que algunas tienen más tiempo que otras. El paso del tiempo hace que el teñido que se hace sobre los cuerpos se vaya perdiendo, haciendo a la muestra más clara.

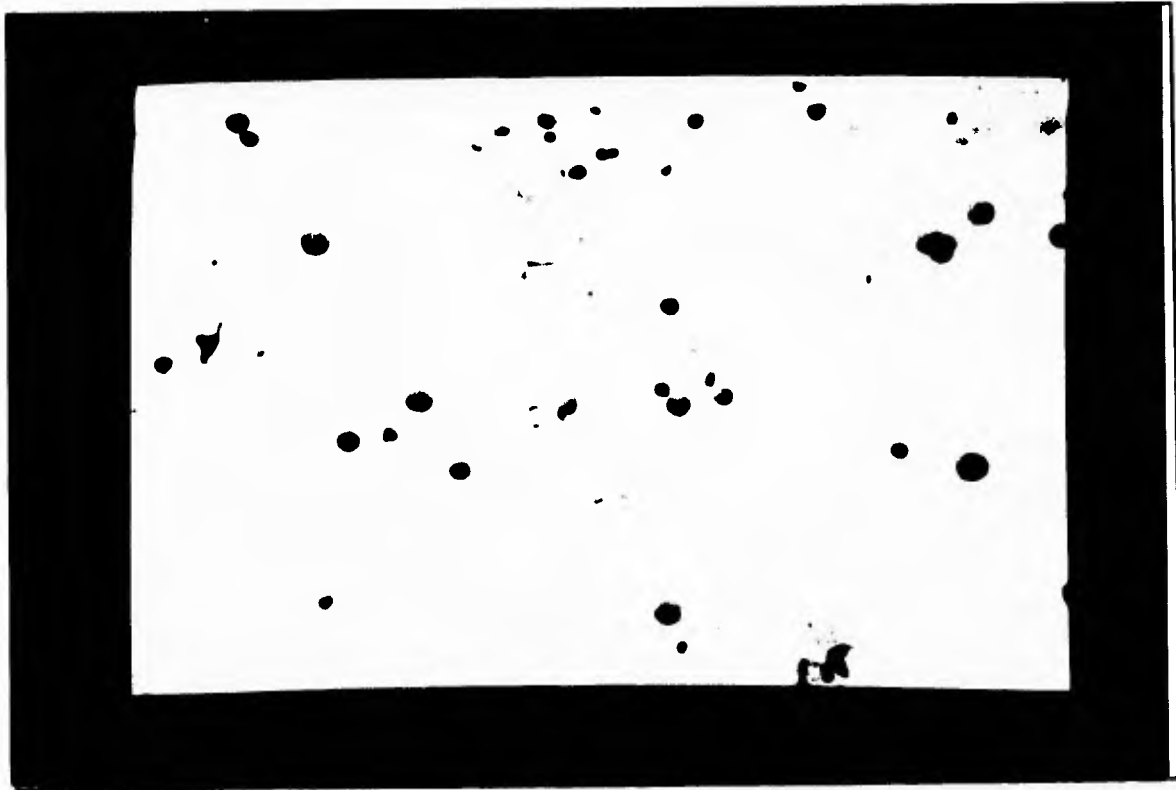


a)

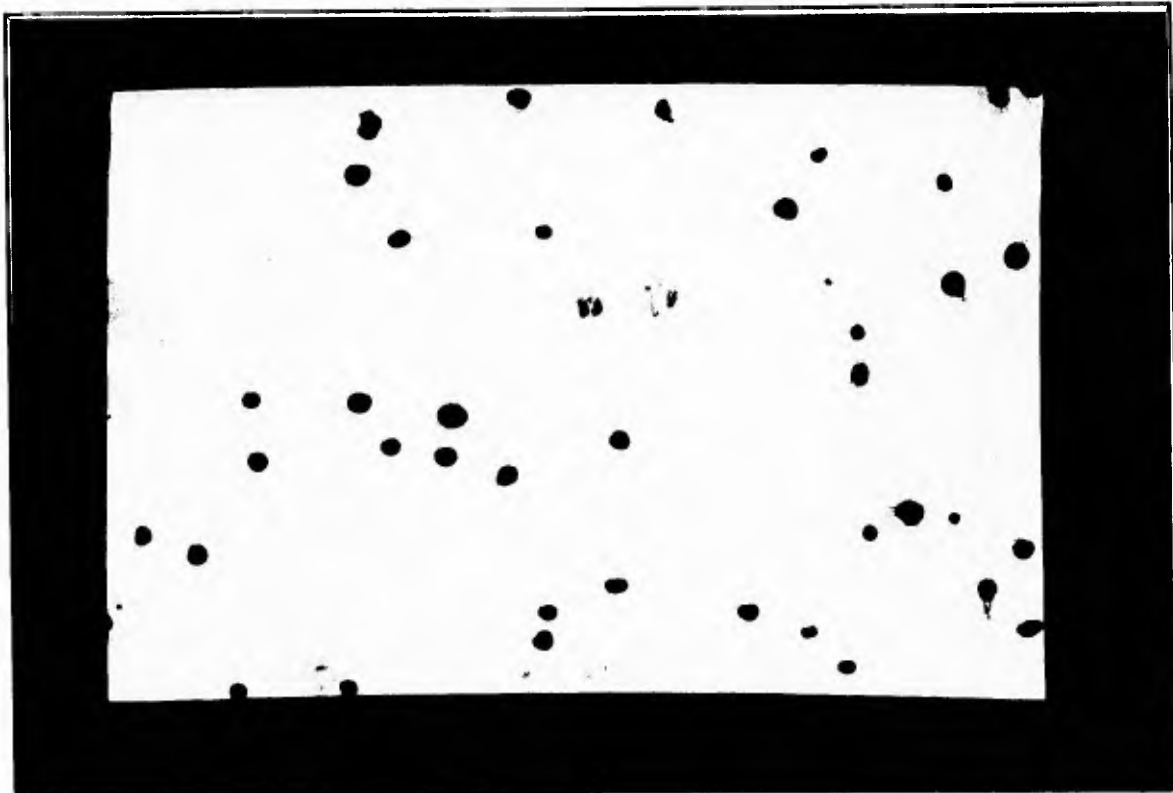


b)

Fig. 5.5. Tipos de laminillas. a) Sucia. b) Limpia.
c) Clara. d) Oscura.



c)



d)

Fig. 5.5. Tipos de laminillas. a) Sucia. b) Limpia.
c) Clara. d) Oscura.

Para resolver este problema se han propuesto dos acciones, la primera es hacer que los biólogos trabajen con preparaciones de la misma edad y, segunda, que la forma en que se preparen sea lo más parecida posible.

La forma en que el sistema reconoce, es una de sus restricciones, debida a la complejidad que requeriría el llevar a cabo el reconocimiento de los objetos de manera inteligente y que al mismo tiempo realizar todas las otras tareas que se requieren para el cálculo del índice mitótico. Sin embargo, esta opción no está descartada para su implementación futura.

V.9. Memoria

Otro problema radica en la creación del espacio de memoria para alojar a las matrices que pudieran usarse para las operaciones de textura. Este espacio puede ser reservado de antemano, lo cual ocasionaría que el programa creciera y quizás tuviera problemas con el límite de los 640 Kb, pero habría que realizar otra investigación para encontrar las dimensiones típicas de las matrices para reservar memoria. Otra posibilidad es el uso de memoria dinámica. De nuevo se tienen problemas de restricción de memoria, lo cual puede resolverse usando archivos; pero es extremadamente lento; o bien, usando la memoria extendida.

Para aprovechar la memoria extendida de las PC se debe crear el manejador de dicha memoria o adecuar los que ya existen en algunos servidores anónimos sobre la red. Para abandonar este tópico se debe tener en cuenta la velocidad del proceso, pues no debe disminuir mucho para que el rendimiento del programa permanezca en los niveles actuales.

CAPITULO VI

Resultados, Perspectivas y Conclusiones.

Se debe ser muy consciente de que el sistema que se propone en el trabajo es un prototipo con el cual se pretendía demostrar que el sistema para el cálculo del índice mitótico era factible de ser realizado. Aunque ya fue probado dentro del Laboratorio de Procesamiento de Imágenes del Centro de Instrumentos todavía falta que sea llevado al Laboratorio de Citogenética del Instituto de Investigaciones Biomédicas para su siguiente etapa de evaluación.

Existen varias vertientes para las siguientes etapas del sistema pero básicamente las tareas a realizar se pueden encerrar en:

a) Mejoramiento de los parámetros descriptores de textura.

Una primera opción es ver qué tan efectivamente se puede describir la textura de un objeto mejorando las mediciones del número de cambios dentro de él. Para esto se propone la medición de los cambios por capas.

Otra vertiente para el análisis de las texturas de las mitosis es la de implementar la medición de textura por medio de matrices de co-ocurrencia (4,5). Esta es una medición de la matriz de co-ocurrencia (dependencia espacial de los niveles de gris) es una medida estadística de segundo orden de la variación de una imagen. Sin embargo, existen varias dificultades para su implementación en este sistema. La primera gran dificultad radica en la complejidad de los cálculos para formar a las matrices y la medición escalar además de la selección de una relación espacial apropiada para formar las matrices. La creación de las matrices se lleva a cabo por medio de la relación de un pixel con sus vecinos; puede hacerse una relación en todos los sentidos, pero esto llevaría mayor tiempo de cálculo.

b) Mejoramiento de los filtros discriminantes.

La mejora de los filtros se refiere a encontrar el punto óptimo para cada una de las desviaciones estándar de cada uno de los parámetros que caracterizan a un objeto, para lograr el mejor de los filtros para cada categoría.

c) Mejoramiento de las opciones del sistema.

Quedan por hacer algunas mejoras en la implementación del sistema en cuanto a su interfaz con el usuario, como ejemplo basta mencionar la creación de archivos de control más flexible.

El objetivo principal del proyecto era el de llevar a cabo un sistema automatizado para la evaluación cuantitativa de la proliferación linfocitaria. Hasta ahora el objetivo se ha

cumplido, obviamente falta hacer las mejoras al sistema para que este pueda funcionar al cien por ciento.

Una de las conclusiones más importantes del trabajo de investigación es que es posible realizar sistemas que tecnológicamente están a nivel internacional. Se debe reconocer que el actual sistema es el primero de su tipo en el mundo. Anteriormente se desarrollaron sistemas que auxilian al cálculo del índice mitótico (3, 16, 17, 18, 19, 20) sin embargo ninguno permite su cálculo automáticamente.

La realización de este sistema ofrece la posibilidad de abrir la puerta a los grupos de investigación interdisciplinaria en el area biomédica, que en México son muy escasos. La aportación que las ingenierías pueden otorgar a las otras áreas no es de servicio, sino de real colaboración para el desarrollo de la tecnología en el país.

Como ejemplo de lo anterior se puede mencionar que el proyecto de investigación MIT 2000 recibió el Premio CANIFARMA 1994 de la Cámara Nacional de la Industria Farmacéutica. Los miembros del equipo interdisciplinario que ganó este premio son: Dr. Gabriel Corkidi Blanco y Edgar Garduño Angeles del Centro de Instrumentos; la M.C. María Garza Jinich, del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), y los doctores Emilio Rojas del Castillo, Regina Montero Montoya y Patricia Ostrosky Weegman, del Instituto de Investigaciones Biomédicas, todos centros de investigación de la Universidad Nacional Autónoma de México.

REFERENCIAS.

1. Otsu, Nobuyuki. "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-9, No 1, Enero de 1979. pp 62-66.
2. Sahoo, P.K., Soltani, S. y Wong, A.K.C. "A Survey of Thresholding Techniques", Computer Vision, Graphics and Image Processing. Vol. 41, 1988. pp 233-245.
3. Garza, M., Rodriguez C., Corkidi G., Montero R., Rojas E., Ostrosky P., Microcomputer-based Supervised System for Automatic Scoring of Mitotic Index in Cytotoxicity Studies, in Advances in Machine Vision: Strategies and Applications. Ed. World Scientific Press, Series in Computer Science, Vol 32, Singapore, 1992, pp. 301-313.
4. Schalkoff, Robert J. Digital Image Processing and Computer Vision. Ed. Wiley, Singapore. 1989. pp 271-275.
5. Haralick, Shapiro. Computer and Robot Vision. Ed. Addison-Wesley, Vol. 1, Estados Unidos. 1992. pp 453-469.
6. Gonzalez, Rafael y Wintz, Paul. Digital Image Processing, Ed. Addison-Wesley, Estados Unidos. segunda edición. 1987.
7. Jähne, Bernd. Digital Image Processing: Concepts, Algorithms and Scientific Applications, Ed. Springer-Verlag, Alemania, Segunda edición. 1991.
8. Russ, John C. Computer-Assisted Microscopy: The measurement and Analysis of Images. Ed. Plenum Press, New York, Estados Unidos. 1990.
9. Microsoft. Microsoft C: Advanced Programming Techniques. Ed. Microsoft, Estados Unidos. 1990.
10. Microsoft. Microsoft FORTRAN: Reference. Ed. Microsoft, Estados Unidos. 1989.
11. Microsoft. Microsoft FORTRAN: Advanced Topics. Ed. Microsoft, Estados Unidos. 1989.
12. MATROX Electronic Systems Limited. PIP Video Digitizer Board for the IBM PC, XT and AT: Hardware Manual. Ed. MATROX Electronic Systems Ltd. Canada. 1988.
13. BIOCUM, Systeme Photometrique D'Analyse D'Images BIOCUM 2000. Ed. BIOCUM, Francia. Versión 1.5
14. U.N.A.M., "Convenio con la Sociedad francesa BIOCUM", Gaceta UNAM. Ed. U.N.A.M., Número 2,443, Enero 25 de 1990. México. pp 3
15. MÄRZHÄUSER GmbH Wetzlar. MultiControl 2000: Documentation. Ed. MÄRZHÄUSER GmbH Wetzlar. Alemania. Noviembre de 1992. Revision 1.0.
16. Green, D.K. and Neurah, P.W. "The design, operation and evaluation of a high speed automatic metaphase finder", The Journal of Histochemistry and Cytochemistry, Vol 22, No 7, 1974. pp 536-545.
17. Johnson, E.T. and Goforth, L.J. "Metaphase spread detection and focus using closed circuit television", The Journal of Histochemistry and Cytochemistry, Vol 22, No 7, 1974. pp 536-545.
18. Preston, Jr. K. and Bartels, P.H. "Automated image processing for cells and tissue", Progress in Medical Imaging, Ed. Springer-Verlag, 1989. Nueva York.

19. Shrivastav, S., Bonar, R. Stone, K. and Paulson, D."An in vitro assay procedure to test chemotherapeutic drugs on cells from human solid tumors", Cancer Research, Vol 40, 1980.pp 4438-4442.
20. Aboussaouira, T., Moustafa, Y. and Idelman, S."Image analysis of cell proliferation in rat thymus throughout development", Thymus, Vol 12, No 1, 1988.pp 167-186.

Apéndice A

Manual de Operación

ACTIVA ENTRENAMIENTO

INSTRUIR ACTIVA

Selecciona algún archivo de entrenamiento para el cálculo del índice mitótico.

FORMATO

INSTRUIR ACTIVA

DESCRIPCION

Este comando permite seleccionar un archivo de entrenamiento de forma interactiva. Los nombres de archivos de entrenamiento son aquellos que tienen terminación ENT y que se hallan en el directorio de trabajo actual.

OBSERVACIONES

Es posible cambiar de directorio de trabajo usando el comando de MSDOS **cd** antecedido por el símbolo **\$**.

EJEMPLOS

INSTRUIR ACTIVA

COMANDOS RELACIONADOS

DEFINE ENTRENAMIENTO

INFORMA ENTRENAMIENTO

PANTALLA

EJECUCION CALCULO DEL INDICE MITOTICO

ACCESO EN MODO DE MENU

En "INSTRU" seleccionar "Activa" y seleccionar de la pantalla el archivo que se desee.

NOTAS

Es posible no elegir algún archivo seleccionando la opción NULL al fondo del menú. Sin embargo, al realizar esto se impide el proceso del cálculo del índice mitótico.

El archivo usado por última ocasión será el que el sistema detecte como activo. Esta información es guardada en el archivo ENTRENA.PAR.

ACTIVAR OBJETIVO

OBJETIVO ACTIVA

Activa algún tipo de objetivo.

FORMATO

OBJETIVO ACTIVA

DESCRIPCION

Activa el tipo de desplazamientos de las platinas para el objetivo que se este usando en el microscopio. Este desplazamiento es necesario para que las platinas cambien de campo de observación según el tipo de objetivo usado en el microscopio.

OBSERVACIONES

Existe un archivo en el cual están definidos los desplazamientos de las platinas según objetivo y se llama OBJETIV.MIC. Para llevar a cabo esta operación es necesario que OBJETIV.MIC este en el directorio de trabajo.

EJEMPLOS

OBJETIVO ACTIVA

COMANDOS RELACIONADOS

OBJETIVO DEFINIR

DEFINE ENTRENAMIENTO

ACTIVA ENTRENAMIENTO

INFORMA ENTRENAMIENTO

EJECUCION DEL CALCULO DEL INDICE MITOTICO

ACCESO EN MODO DE MENU

En "OBJETIVO" seleccionar "Activar" y seleccionar de la pantalla el objetivo que se desee.

NOTAS

Si no ha sido seleccionado un desplazamiento de las platinas entonces no se podrá ejecutar el cálculo del índice mitótico.

DEFINIR BARRIDO INDIVIDUAL

BARRIDO DEFINIR INDIV

Define la zona de barrido en cada laminilla instalada en las platinas.

FORMATO

BARRIDO DEFINIR INDIV

DESCRIPCION

Por medio de esta rutina es posible definir la zona de interés para cada laminilla instalada en las platinas. Se debe recordar que el barrido de las laminillas siempre se realiza en zigzag dentro de una zona rectangular.

OBSERVACIONES

El barrido individual sirve para cuando el usuario no desea usar el barrido predefinido que el sistema usa para cada laminilla. Debe ser precisada la zona de barrido para cada laminilla que se halle instalada en las platinas.

La definición de la zona de barrido general se hace de manera interactiva señalando las esquinas superior izquierda e inferior derecha del rectángulo de barrido dentro de cada laminilla.

EJEMPLOS

BAR DEF IN

BARRRIDO DEFINIR INDIV

COMANDOS RELACIONADOS

BARRIDO NUMPORTA

BARRIDO PREDEF

BARRIDO DEFINIR UNICO

ACCESO EN MODO DE MENU

En la opción "BARRIDO" se elige "Definir". Una vez en "Definir" se selecciona "Individual"

NOTAS

No es necesario definir una zona individual de barrido si la zona predefinida del sistema cubre las necesidades de análisis, sin embargo el sistema si requiere de que exista una zona de barrido definida antes de realizar las operaciones que involucren algún tipo de examen usando las platinas.

También se le debe decir al sistema cuantas laminillas hay instaladas antes de llamar este comando, de otra forma el sistema solo reconoce que hay únicamente una laminilla.

DEFINIR BARRIDO GENERAL

BARRIDO DEFINIR UNICO

Define una zona de barrido común para todas las laminillas colocadas en las platinas.

FORMATO

BARRIDO DEFINIR UNICO

DESCRIPCION

Por medio de esta rutina es posible definir la zona de interés para todas las laminillas instalada en las platinas. Se debe recordar que el barrido de las laminillas siempre se realiza en zigzag dentro de una zona rectangular.

OBSERVACIONES

El barrido general sirve para cuando el usuario no desea usar el barrido predefinido que el sistema usa para todas las laminillas. Debe ser precisada la zona de barrido para solo una laminilla y esta será repetida para todas las laminillas instaladas en el sistema.

La definición de la zona de barrido general se hace de manera interactiva señalando las esquinas superior izquierda e inferior derecha del rectángulo de barrido dentro de una laminilla.

EJEMPLOS

BARRIDO DEFINIR UNICO

BAR DEF UN

COMANDOS RELACIONADOS

BARRIDO NUMPORTA

BARRIDO PREDEF

BARRIDO DEFINIR INDIVIDUAL

ACCESO EN MODO DE MENU

En la opción "BARRIDO" se elige "Definir". Una vez en "Definir" se selecciona "1 para todos"

NOTAS

No es necesario definir una zona general de barrido si la zona predefinida del sistema cubre las necesidades de análisis, sin embargo el sistema si requiere de que exista una zona de barrido definida antes de realizar las operaciones que involucren algún tipo de examen usando las platinas.

También se le debe decir al sistema cuantas laminillas hay instaladas antes de llamar este comando, de otra forma el sistema solo reconoce que hay únicamente una laminilla.

DEFINIR BARRIDO PREDEFINIDO

BARRIDO PREDEFINIDO

Ordena al sistema a usar el barrido de las platinas predefinido.

FORMATO

BARRIDO PREDEF

DESCRIPCION

El sistema tiene definida las zonas de interés típicas para todas las laminillas. Llamando este comando se pide al sistema que use estas zonas de interés para el análisis de la platinas.

OBSERVACIONES

Cuando se declara el número de laminillas que se hallan instaladas en las platinas el sistema el sistema, por principio, auto declara que usará las zonas predefinidas. Si el usuario desea usar otras debe llamar a las rutinas de definición.

La definición de la zona de barrido se hace automáticamente.

EJEMPLOS

BARRIDO PREDEF

BAR PRE

COMANDOS RELACIONADOS

BARRIDO NUMPORTA

BARRIDO DEFINIR INDIVIDUAL

BARRIDO DEFINIR UNICO

ACCESO EN MODO DE MENU

En la opción "BARRIDO" se elige "Predefinido".

NOTAS

Las zonas predefinidas de barrido siempre son declaradas al definir el número de laminillas a examinar.

El sistema al entrar declara que se usará una laminilla con zona predefinida.

DEFINIR BARRIDO PREDEFINIDO

BARRIDO PREDEFINIDO

Ordena al sistema a usar el barrido de las platinas predefinido.

FORMATO

BARRIDO PREDEF

DESCRIPCION

El sistema tiene definida las zonas de interés típicas para todas las laminillas. Llamando este comando se pide al sistema que use estas zonas de interés para el análisis de la platinas.

OBSERVACIONES

Cuando se declara el número de laminillas que se hallan instaladas en las platinas el sistema el sistema, por principio, auto declara que usará las zonas predefinidas. Si el usuario desea usar otras debe llamar a las rutinas de definición.

La definición de la zona de barrido se hace automáticamente.

EJEMPLOS

BARRIDO PREDEF

BAR PRE

COMANDOS RELACIONADOS

BARRIDO NUMPORTA

BARRIDO DEFINIR INDIVIDUAL

BARRIDO DEFINIR UNICO

ACCESO EN MODO DE MENU

En la opción "BARRIDO" se elige "Predefinido".

NOTAS

Las zonas predefinidas de barrido siempre son declaradas al definir el número de laminillas a examinar.

El sistema al entrar declara que se usará una laminilla con zona predefinida.

DEFINIR NUMERO DE PORTAOBJETOS

BARRIDO DEFINE NUMERO DE PORTAOBJETOS

Declara al sistema el número de laminillas que serán analizadas por el sistema.

FORMATO

BARRIDO NUMPORTA n

n = número de laminillas instaladas en el sistema para su análisis. Se debe poner un número entero entre 1 y 8.

DESCRIPCION

Por medio de esta rutina es posible declarar el número de laminillas que el sistema deberá analizar.

OBSERVACIONES

Es importante que se declare el número de laminillas a usar antes de cada comando que involucre el análisis de las muestras que se hallen en las platinas sino el sistema solo analizará una laminilla.

Si se desea usar otro tipo de barrido diferente al que el sistema usa por principio es necesario definirlo con las rutinas UNICO e INDIV.

EJEMPLOS

BAR NUM 5 Le dice al sistema que hay 5 laminillas para analizar.

BAR NUM 8 Le dice al sistema que hay ocho laminillas para analizar.

COMANDOS RELACIONADOS

BARRIDO DEFINIR INDIVIDUAL

BARRIDO DEFINIR UNICO

BARRIDO PREDEF

EJECUCION DEL CALCULO DEL INDICE MITOTICO

ACCESO EN MODO DE MENU

En la opción "BARRIDO" del menú de "MITOSIS" seleccionar "Nº Porta Obj"

EL sistema despliega: 'Cantidad de porta objetos?'. Se debe darle un numero entero entre 1 y 8.

NOTAS

Siempre es necesario declarar el número de laminillas a analizar antes de ejecutar los comandos de cálculo del índice mitótico.

Siempre que se declara el número de laminillas posteriormente el sistema define las zonas de barrido con aquellas predefinidas.

DEFINIR OBJETIVO

OBJETIVO DEFINIR

Define el tipo de objetivo a ser usado y los desplazamientos de las platinas para dicho objetivo.

FORMATO

OBJETIVO DEFINIR

DESCRIPCION

Con este comando es posible relacionar el tipo de objetivo con el desplazamiento necesario que deben realizar las platinas para que coincidan los campos de observación contiguos.

OBSERVACIONES

Existe un archivo en el cual están definidos los desplazamientos de las platinas según objetivo y se llama OBJETIV.MIC. Para llevar a cabo esta operación es necesario que OBJETIV.MIC este en el directorio de trabajo.

Al llamar este comando se debe mover un objeto dentro de la pantalla "gráfica" para definir los desplazamientos que debe realizar para el objetivo seleccionado.

EJEMPLOS

OBJETIVO DEFINIR

OBJETIV DEF

COMANDOS RELACIONADOS

OBJETIVO ACTIVAR

OVER

ACCESO EN MODO DE MENU

En la opción "OBJETIVO" del menú de "MITOSIS" seleccionar "Definir".

NOTAS

No siempre es necesario definir un objetivo pues puede que ya este definido en el archivo OBJETIV.MIC. Si es así tan solo se requiere llamar al comando **OBJETIVO ACTIVAR**.

Al entrar al sistema es verificado el último objetivo usado. Este será el que será activado por definición.

CREAR NUEVO ENTRENAMIENTO

INSTRUIR NUEVO

Con este comando es posible crear un nuevo archivo de entrenamiento para el cálculo del índice mitótico.

FORMATO

INSTRUIR NUEVO

DESCRIPCION

Al usar este comando se pide al sistema que comience la creación de un nuevo archivo de entrenamiento.

La creación del nuevo archivo es de manera interactiva requiere que el usuario de el nombre del nuevo archivo. Si este archivo existe preguntará si se sobrescribe o se usa otro nombre.

OBSERVACIONES

Para saber como entrenar el sistema por favor ver sección de entrenamiento en el manual de protocolos.

Al terminar de crear el nuevo entrenamiento este archivo será activado como archivo de entrenamiento actual.

El número de cuerpos que se muestrean usando esta rutina será responsabilidad del usuario.

EJEMPLOS

INSTRUIR NUEVO

INST NUE

COMANDOS RELACIONADOS

INSTRUIR INFORMA

INSTRUIR ACTIVA

EJECCION CALCULO DEL INDICE MITOTICO

ACCESO EN MODO DE MENU

En la opción "INSTRUIR" del menú "MITOSIS" seleccionar "Nuevo"

NOTAS

Es necesario tener activo un archivo de entrenamiento antes de pedir el comando que calcula el índice mitótico.

Siempre usar un entrenamiento que cumpla con las condiciones que serán usadas en la próxima ejecución del cálculo del índice mitótico.

Modifica el archivo ENTRENA.PAR.

EJECUTA INDICE MITOTICO

EJECUCION INDICE MITOTICO

Permite la ejecución del cálculo del índice mitótico.

FORMATO

EJECUCION INDICE

DESCRIPCION

Por medio del uso de este comando el sistema comienza la ejecución del cálculo del índice mitótico en las laminillas que se hayan declarado instaladas.

OBSERVACIONES

La definición del número de controles y experimentos asociados se hace al llamar a esta rutina.

El sistema usa el archivo NUCLEOS.REC para vaciar los resultados del cálculo del índice mitótico. Para salvar este archivo se puede usar el comando de MSDOS **rename** de las siguiente forma: \$ ren nucleos.rec nombre.rec

Es importante usar la extensión REC pues la rutina de verificación busca solo ese tipo de archivos.

La ejecución del índice mitótico puede ser interrumpida oprimiendo la tecla Esc.

EJEMPLOS

EJECU INDICE

COMANDOS RELACIONADOS

EJECUCION CMD

EJECUCION MANDI

CONTEO

INSTRUIR ACTIVA

INSTRUIR NUEVO

BARRIDO NUMPORTA

BARRIDO DEFINIR INDIVIDUAL

BARRIDO DEFINIR UNICO

BARRIDO PREDEFINIDO

OBJETIVO ACTIVAR

OBJETIVO DEFINIR

OVER

PORCENT

ACCESO EN MODO DE MENU

En la opción "EJECUCION" del menú de "MITOSIS" seleccionar "Indice Mit"

NOTAS

Antes de llamar este comando declarar el número de laminillas que están instaladas y el tipo de barrido que se usará.

EJECUTA BARRIDO Y ARCHIVO CMD

EJECUCION CMD

Permite la ejecución de un archivo de comandos CMD sobre cada campo del barrido del índice mitótico.

FORMATO

EJECUCION CMD

DESCRIPCION

Por medio del uso de este comando es posible ejecutar un archivo de comandos CMD sobre cada campo que es analizado en el barrido del índice mitótico.

OBSERVACIONES

El orden en que las laminillas serán exploradas es dado al llamar este comando.

El tipo de barrido que se llevará a cabo sobre cada laminilla debe ser declarado antes de llamara este comando.

El comando comprueba la existencia del archivo CMD en el directorio de trabajo.

EJEMPLOS

EJECUCION CMD

EJE CM

COMANDOS RELACIONADOS

EJECUCION INICE

EJECUCION MANDI

BARRIDO NUMPORTA

BARRIDO DEFINIR INDIVIDUAL

BARRIDO DEFINIR UNICO

BARRIDO PREDEFINIDO

OBJETIVO ACTIVAR

OBJETIVO DEFINIR

OVER

ACCESO EN MODO DE MENU

En la opción "EJECUCION" del menú de "MITOSIS" seleccionar "Lee CMD"

NOTAS

Antes de llamar este comando declarar el número de laminillas que están instaladas y el tipo de barrido que se usará.

No se puede ejecutar comandos del modulo MITOSIS.

EJECUTA CALCULO INDICE MITOTICO Y ARCHIVO CMD

EJECUCION DE ARCHIVO CMD Y CALCULO DEL INDICE MITOTICO

Este comando permite la ejecución de un archivo de comandos CMD en cada campo de análisis antes de ejecutar el reconocimiento de los objetos para el cálculo del índice mitótico.

FORMATO

EJECUCION CANDI

DESCRIPCION

Por medio del uso de este comando es posible ejecutar un archivo de comandos CMD sobre cada campo de análisis antes de llevar a cabo el reconocimiento de los objetos que es necesario para calcular el índice mitótico.

OBSERVACIONES

La definición del número de controles y experimentos asociados se hace al llamar a esta rutina.

El sistema usa el archivo NUCCMD.REC para vaciar los resultados del cálculo del índice mitótico. Para salvar este archivo se puede usar el comando de MSDOS rename de las siguiente forma: \$ ren nuccmd.rec nombre.rec

Es importante usar la extensión REC pues la rutina de verificación busca solo ese tipo de archivos.

El comando comprueba la existencia del archivo CMD en el directorio de trabajo.

EJEMPLOS

EJECUCION CANDI

EJE CAN

COMANDOS RELACIONADOS

EJECUCION CMD

EJECUCION MANDI

CONTEO

INSTRUIR ACTIVA

INSTRUIR NUEVO

BARRIDO NUMPORTA

BARRIDO DEFINIR INDIVIDUAL

BARRIDO DEFINIR UNICO

BARRIDO PREDEFINIDO

OBJETIVO ACTIVAR

OBJETIVO DEFINIR

OVER

PORCENT

ACCESO EN MODO DE MENU

En la opción "EJECUCION" del menú de "MITOSIS" seleccionar "Lee CMD&Mit"

NOTAS

Antes de llamar este comando declarar el número de laminillas que están instaladas y el tipo de barrido que se usará.

No se puede ejecutar comandos del modulo MITOSIS.

ENFOCA FINO

ENFOQUE FINO

Esta rutina permite hacer la operación de enfoque fino en el campo de observación actual.

FORMATO

ENFOQUE FINO

DESCRIPCION

Este comando permite realizar le enfoque de la imagen de forma fina en el campo que se observa en la pantalla gráfica.

OBSERVACIONES

Si la adquisición de las imágenes se encuentra en modo congelado al llamar este comando pasa a ser modo continuo.

EJEMPLOS

ENF FIN

COMANDOS RELACIONADOS

ENFOQUE GRUESO

MOV

POS

ACCESO EN MODO DE MENU

En la opción "ENFOQUE" del menú de "MITOSIS" se selecciona "Fino".

NOTAS

El barrido de enfoque es limitado. Para enfocar un campo muy borroso es preferible usar una combinación de enfoque fino y enfoque grueso.

ENFOCA GRUESO

ENFOQUE GRUESO

Esta rutina permite hacer la operación de enfoque fino en el campo de observación actual.

FORMATO

ENFOQUE GRUESO

DESCRIPCION

Este comando permite realizar le enfoque de la imagen de forma gruesa en el campo que se observa en la pantalla gráfica.

OBSERVACIONES

Si la adquisición de las imágenes se encuentra en modo congelado al llamar este comando pasa a ser modo continuo.

EJEMPLOS

ENF GR

COMANDOS RELACIONADOS

ENFOQUE FINO

MOV

POS

ACCESO EN MODO DE MENU

En la opción "ENFOQUE" del menú de "MITOSIS" se selecciona "Grueso".

NOTAS

El barrido de enfoque es limitado. Para enfocar un campo muy borroso es preferible usar una combinación de enfoques gruesos.

AFINA IMAGEN

SHARP

Realiza el afinamiento de la imagen.

FORMATO

AFIL

DESCRIPCION

Realiza el afinamiento de la imagen usando la rutina `fg_sh2` de la tarjeta MATROX.

OBSERVACIONES

Se usa el siguiente tipo de máscara:

-1	-1	-1
-1	9	-1
-1	-1	-1

EJEMPLOS

ACQ BRU

AFIL

SEUIL

COMANDOS RELACIONADOS

REHAUSMENT

LISSAGE

ACCESO EN MODO DE MENU

Seleccionar la opción "SHARP" del menú de "MITOSIS".

NOTAS

Esta operación aumenta el contraste de la imagen por lo que puede ocasionar saturación en el número de contornos.

INFORMA ENTRENAMIENTO

INSTRUIR INFO

Con este comando es posible saber el entrenamiento que esta activo.

FORMATO

INSTRUIR INFO

DESCRIPCION

Enseña el nombre del archivo de entrenamiento que se encuentra activo.

OBSERVACIONES

EJEMPLOS

INSTRUIR ACTIV

INSTRUIR INFO

COMANDOS RELACIONADOS

INSTRUIR ACTIVA

INSTRUIR DEFINIR

ACCESO EN MODO DE MENU

Dentro de la opción "INSTRUIR" del menú de "MITOSIS" seleccionar la opción "Info".

NOTAS

DEFINE NUMERO DE CUERPOS

CONTEO

Rutina para definir el número máximo de objetos a contar en el índice mitótico.

FORMATO

CONTEO n

n = Número, entero entre 1 y 2'000,000, de objetos a contar.

DESCRIPCION

En el cálculo del índice mitótico hay dos criterios para detener el conteo en una laminilla. El primero es que se alcanza el número máximo de objetos a contar y, segundo, se termino la zona de barrido.

OBSERVACIONES

Siempre debe definirse el número de objetos antes de ejecutar el cálculo del índice mitótico.

EJEMPLOS

CONTEO 1000 Se contarán mil objetos en cada laminilla.

CON 200 Se contarán dos cientos objetos en cada laminilla.

COMANDOS RELACIONADOS

EJECUCION INDICE

EJECUCION CANDI

ACCESO EN MODO DE MENU

Seleccionar "CONTEO" del menú de "MITOSIS".

Aparecerá el letrero 'Cantidad de cuerpos a contar:?' después del cual se debe introducir el número de objetos que se desean contar.

NOTAS

Por definición el programa contará dos mil objetos.

VERIFICACION

VERIFICA

Verifica la ejecución del índice mitótico.

FORMATO

VERIFICA

DESCRIPCION

Por medio de este comando es posible seleccionar algún archivo REC en el cual se guardo la bitácora de la ejecución del cálculo del índice mitótico.

OBSERVACIONES

Los archivos REC deben estar en el directorio de trabajo actual.

EJEMPLOS

EJE IND

VERI

COMANDOS RELACIONADOS

EJECUCION INDICE

EJECUCION CANDI

ACCESO EN MODO DE MENU

Seleccionar la opción "VERIFICA" del menú "MITOSIS".

NOTAS

Si se quiere revisar alguna ejecución del índice mitótico que fue interrumpida entonces habrá problemas en la revisión de este archivo.

DESPLAZAMIENTO EN X

DESPX

Este comando define el desplazamiento en X y máximo movimiento en X de las platinas.

FORMATO

DESPX

DESCRIPCION

Este comando permite definir el desplazamiento en X para cambiar de portaobjetos y desplazamiento en X del barrido.

OBSERVACIONES

Modifica el archivo DESP.PAR.

EJEMPLOS

DESPX

COMANDOS RELACIONADOS

OVER

PORCY

OBJETIVO DEFINIR

ACCESO EN MODO DE MENU

Seleccionar la opción "DESP EN X" del menú "MITOSIS".

NOTAS

Si el archivo DESP.PAR no se encuentra el sistema no puede ejecutar el cálculo del índice mitótico. Se debe definir este desplazamiento al entrar al sistema.

PORCENTAJE EN Y

PORCY

Este comando ayuda definir el porcentaje de las laminillas que se usará para el barrido predefinido.

FORMATO

PORCY p

p = es el porcentaje de la distancia del eje Y que será usado para el barrido predefinido.

Es un número real entre 0 y 100.

DESCRIPCION

El comando **DESPY** ayuda definir el porcentaje de las laminillas que se usará para el barrido predefinido.

OBSERVACIONES

El desplazamiento en Y debe de ser declarado antes de definir un barrido.

El porcentaje de Y se encuentra declarado en el archivo **DESP.PAR** y por ende al iniciar el sistema esta cantidad es leída para no tener que volver a definir el porcentaje cada vez que se usa el sistema.

EJEMPLOS

DESPX

PORCY

COMANDOS RELACIONADOS

OBJETIVO DEFINIR

OBJETIVO ACTIVAR

OVER

DESPX

ACCESO EN MODO DE MENU

Seleccionar la opción "**PORCENT Y**" del menú "**MITOSIS**". Aparecerá un letrero que dice 'Porcentaje (entero positivo)?' y después del cual debe introducirse el número real entre cero y cien.

NOTAS

El archivo **DESP.PAR** será creado al modificar este valor.

PRUEBA POR PANTALLA

SCR

Comando que reconoce núcleos estimulados, mitosis cerradas y mitosis abiertas en una imagen.

FORMATO

SCR

DESCRIPCION

Comando que reconoce núcleos estimulados, mitosis cerradas y mitosis abiertas en la imagen que se encuentra en la pantalla "gráfica".

OBSERVACIONES

Este comando ayuda a ver el comportamiento del archivo de entrenamiento antes de usar el comando de ejecución del cálculo del índice mitótico.

EJEMPLOS

INSTRUIR NUEVO

SCR

COMANDOS RELACIONADOS

EJECUCION INDICE

INSTRUIR ACTIVA

INSTRUIR NUEVO

ACCESO EN MODO DE MENU

Seleccionar la opción "PANT" del menú "MITOSIS".

NOTAS

La imagen no se renueva al llamar este comando así que se debe tener cuidado de usar la imagen adecuada al llamar este comando.

POSICION

POSICION

Este comando muestra la posición en que se encuentran las platinas.

FORMATO

POS

DESCRIPCION

Este comando proporciona la posición de las platinas en los tres ejes coordenados.

OBSERVACIONES

El comando siempre regresa tres números del tipo real representando la posición en X,Y y Z respectivamente.

EJEMPLOS

MOV 20 40 0

POS

COMANDOS RELACIONADOS

MOV

ENFOQUE FINO

ENFOQUE GRUESO

ACCESO EN MODO DE MENU

NOTAS

PORCENTAJE DE INDICE MITOTICO

PORCENT

Modifica el porcentaje del índice mitótico mínimo para los controles.

FORMATO

PORCENT p

p = es un número entre cero y cien que representa el porcentaje mínimo para que las laminillas "controles" sean clasificadas como buenos.

DESCRIPCION

El comando permite definir el porcentaje mínimo del índice mitótico en las laminillas "control".

OBSERVACIONES

Si en las laminillas "control" el índice mitótico es igual o mayor a este porcentaje entonces se revisan sus laminillas "experimento".

EJEMPLOS

PORCENT 10

CONTEO 2000

BAR NUM 6

EJE IND

COMANDOS RELACIONADOS

CONTEO

BARRIDO NUMPORTA

BARRIDO PREDEF

BARRIDO DEFINIR INDIV

BARRIDO DEFINIR UNICO

EJECUCION INDICE

EJECUCION CANDI

ACCESO EN MODO DE MENU

NO HAY

NOTAS

Al entrar al sistema este porcentaje es igual al 5%.

INFORMA CUERPOS A CONTAR

INFORMA CUERPOS A CONTAR

Este comando informa el número de objetos que serán contados al realizar el cálculo del índice mitótico.

FORMATO

INFCON

DESCRIPCION

Imprime el número máximo de cuerpos a ser contados en cada laminilla.

OBSERVACIONES

Este es uno de los criterios para detener el análisis de cada laminilla, el otro criterio es acabar la zona de barrido.

EJEMPLOS

CONTEO 2000

INFCON

COMANDOS RELACIONADOS

CONTEO

EJECUCION INDICE

EJECUCION CANDI

ACCESO EN MODO DE MENU

NO HAY

NOTAS

MOVER

MOVER

Este comando permite mover las platinas a una posición específica.

FORMATO

MOVER x y z tipo

x = desplazamiento en el eje X.

y = desplazamiento en el eje Y.

z = desplazamiento en el eje Z.

tipo = define el tipo de desplazamiento. Si tipo = 0 entonces el movimiento es absoluto.

Si tipo = 1 entonces el movimiento es relativo.

DESCRIPCION

Permite moverse por el espacio (x,y,z) de las platinas.

OBSERVACIONES

Es importante definir el tipo de desplazamiento.

EJEMPLOS

MOV 10 10 0 0 Las platinas se mueven a la posición absoluta (10,10,0)

MOV 10 10 0 1 Las platinas se mueven a la posición
(actual x +10, actual y + 10, actual z + 0).

COMANDOS RELACIONADOS

POS

ACCESO EN MODO DE MENU

NO HAY

NOTAS

DEFINE TRASLAPE

OVERLAP

Permite definir el traslape en (x,y) que será usada para cada toma.

FORMATO

OVER

DESCRIPCION

Este comando define el traslape que será usado en (x,y) para que algún objeto que quede en medio de dos pantallas, debido al desplazamiento de las platinas para el objetivo actual, pueda quedar entero en la pantalla.

OBSERVACIONES

Es necesario definir algún desplazamiento según el objetivo que se estén definidos. Normalmente, si al activar un objetivo no hay traslape definido para dicho objetivo entonces el sistema imprime un mensaje de error sobre la ausencia de traslape.

EJEMPLOS

OBJETIVO ACTIVAR

OVER

COMANDOS RELACIONADOS

OBJETIVO DEFINIR

OBJETIVO ACTIVAR

PORCY

DESPX

ACCESO EN MODO DE MENU

NO HAY

NOTAS

Este traslape esta definido en el archivo DESP.PAR. Si DESP.PAR no existe al entrar al sistema se debe definir un traslape.

El traslape debe estar bien definido para que no se cuenten doble algunos objetos y que no falte contar ninguno.

MUESTRA TRASLAPE

SHOW

Muestra el traslape usado en (x,y) para cada toma.

FORMATO

SHOW

DESCRIPCION

Muestra el traslape usado en (x,y) para que cada toma se encime sobre la otra.

OBSERVACIONES

EJEMPLOS

OBJETIVO ACTIVAR

OVER

SHOW

COMANDOS RELACIONADOS

OBJETIVO ACTIVAR

OBJETIVO DEFINIR

DESPX

PORCY

OVER

ACCESO EN MODO DE MENU

NO HAY

NOTAS

SEGMENTA

SEPARA

Realiza un proceso de segmentación excluyendo al contorno de mayor área, afinándola para después sobreponer los contornos antes hallados.

FORMATO

SEPARA

DESCRIPCION

Realiza un proceso de segmentación excluyendo al contorno de mayor área, afinándola para después sobreponer los contornos antes hallados.

OBSERVACIONES

Es otra forma de segmentar a la ya implementada en BIOCOM. Esta permite segmentar automáticamente usando el método de división desarrollado por Otsu.

EJEMPLOS

ACQ BRU

SEPARA

HISTO ZON

COMANDOS RELACIONADOS

SEUIL

REINI

ISOCONTOUR

REAFFICHE

DETRUIT

ACCESO EN MODO DE MENU

NO HAY

NOTAS

Al usar esta rutina se pierden los contornos anteriores a esta operación

PROCOLOS

GENERAL

Al entrar al sistema se tienen definido lo siguientes parametros:

- 2000 objetos a contar en cada laminilla.
- una sola laminilla instalada
- barrido predefinido para dicha laminilla.
- 5% índice mitótico mínimo para examinar los experimentos asociados a los controles.
- lectura de archivos **DESP.PAR**, **OBJETIV.MIC**, **ENTRENA.PAR** y **OVERLAP.PAR**.

ENTRENAMIENTO

Al crear un nuevo entrenamiento se debe dar solo el nombre del nuevo archivo. El entrenamiento se crea seleccionando los campos que contengan núcleos estimulados, cromosomas cerrados y/o cromosomas abiertos.

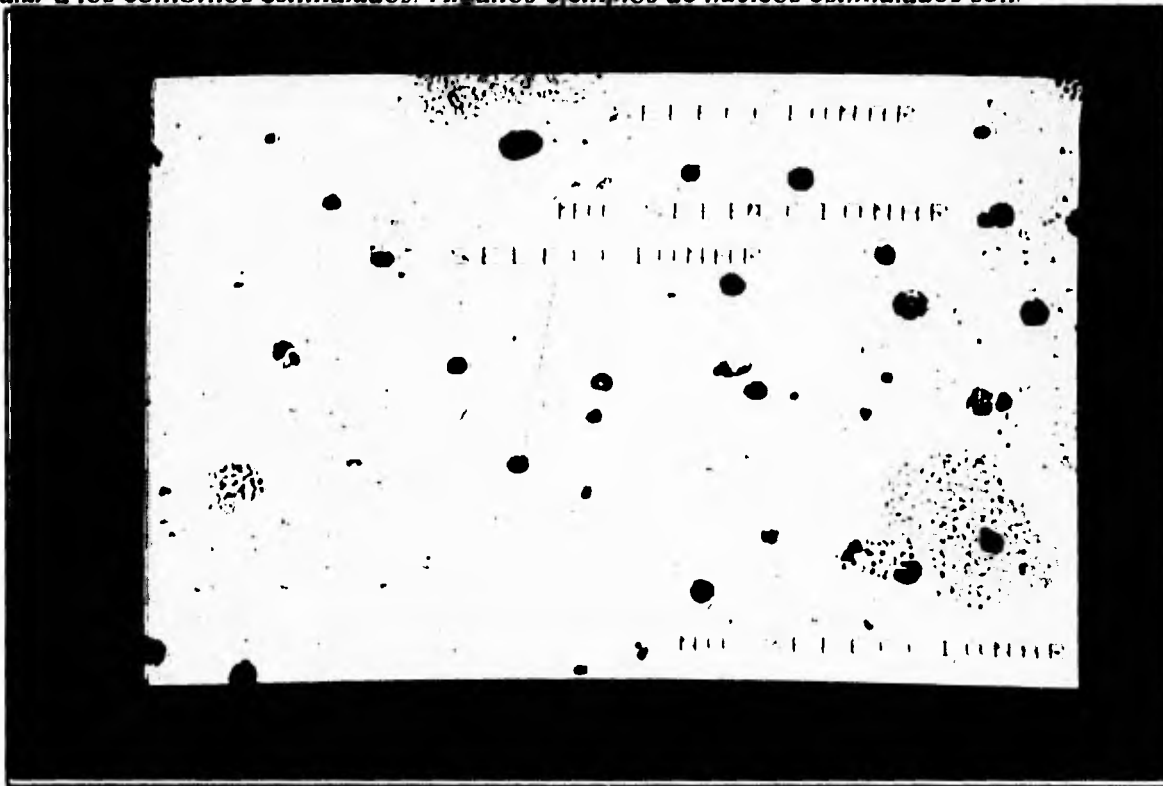
Una vez en un campo que se desee usar como parte del entrenamiento simplemente se activa la operación de selección de cuerpos en dicho campo por medio de la tecla *I* o el botón izquierdo del ratón.

Después de hacer lo anterior, el sistema segmenta la pantalla y espera a que se le señalen los diferentes tipos de cuerpos de interés que se hallan en la pantalla.

El patrón que se ha seguido para elegir los diferentes cuerpos es:

Núcleos

Para seleccionar núcleos estimulados se oprime la tecla *E* antes de usar el ratón para señalar a los contornos estimulados. Algunos ejemplos de núcleos estimulados son:

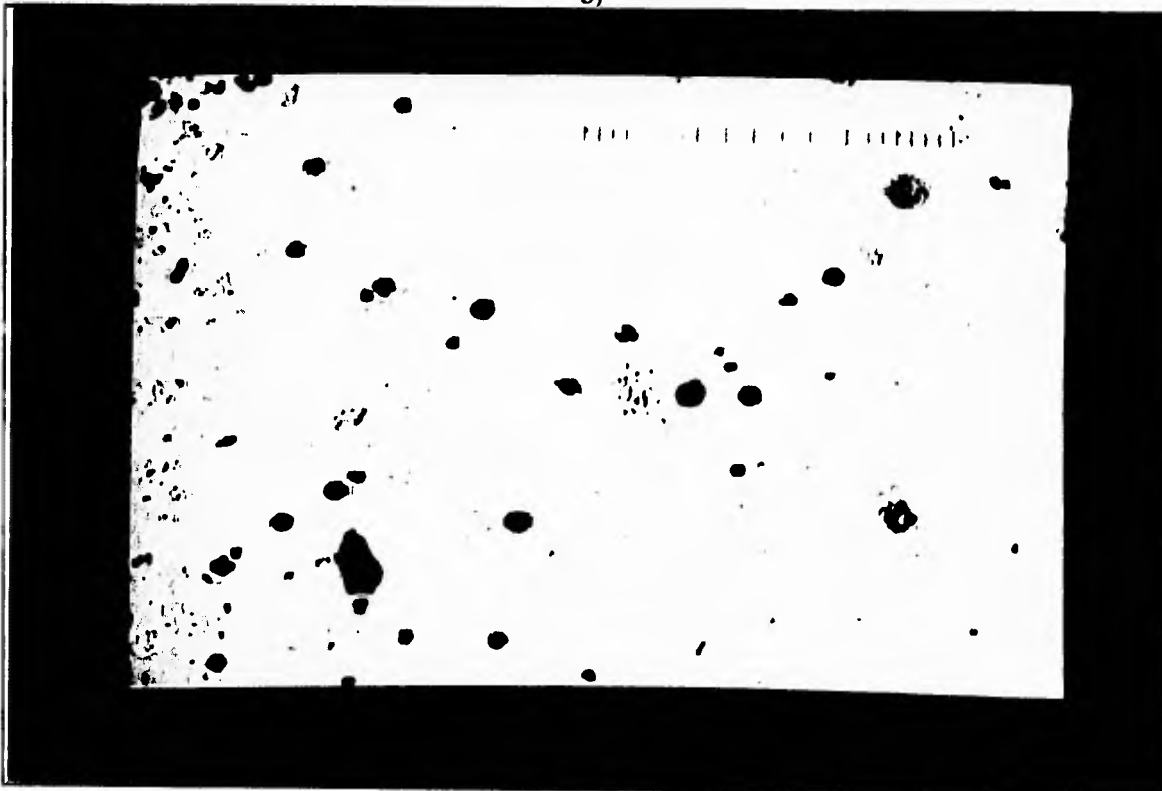


a)

Fig. A.E.1. Núcleos estimulados en el entrenamiento. a) ejemplo 1. b) ejemplo 2. c) ejemplo 3.



b)



c)

Fig. A.E.1. Núcleos estimulados en el entrenamiento. a) ejemplo 1. b) ejemplo 2. c) ejemplo 3.

algunos cuerpos que no se usan como núcleos estimulados son:



Fig. A.E.2 Contornos que no son núcleos estimulados en el entrenamiento.

Mitosis cerradas

Antes de seleccionar un contorno como mitosis cerrada se debe oprimir la tecla *C*.

Quizas el problema más frecuente es decidir cuando se trata de una mitosis cerrada y cuando de una abierta. Se han usado dos criterios para decidir los contornos que representan mitosis cerradas.

1. Que el objeto este rodeado por un solo contorno.
2. Que el contorno tenga una forma más o menos redondeada.

Algunos ejemplos de mitosis cerradas son:



a)



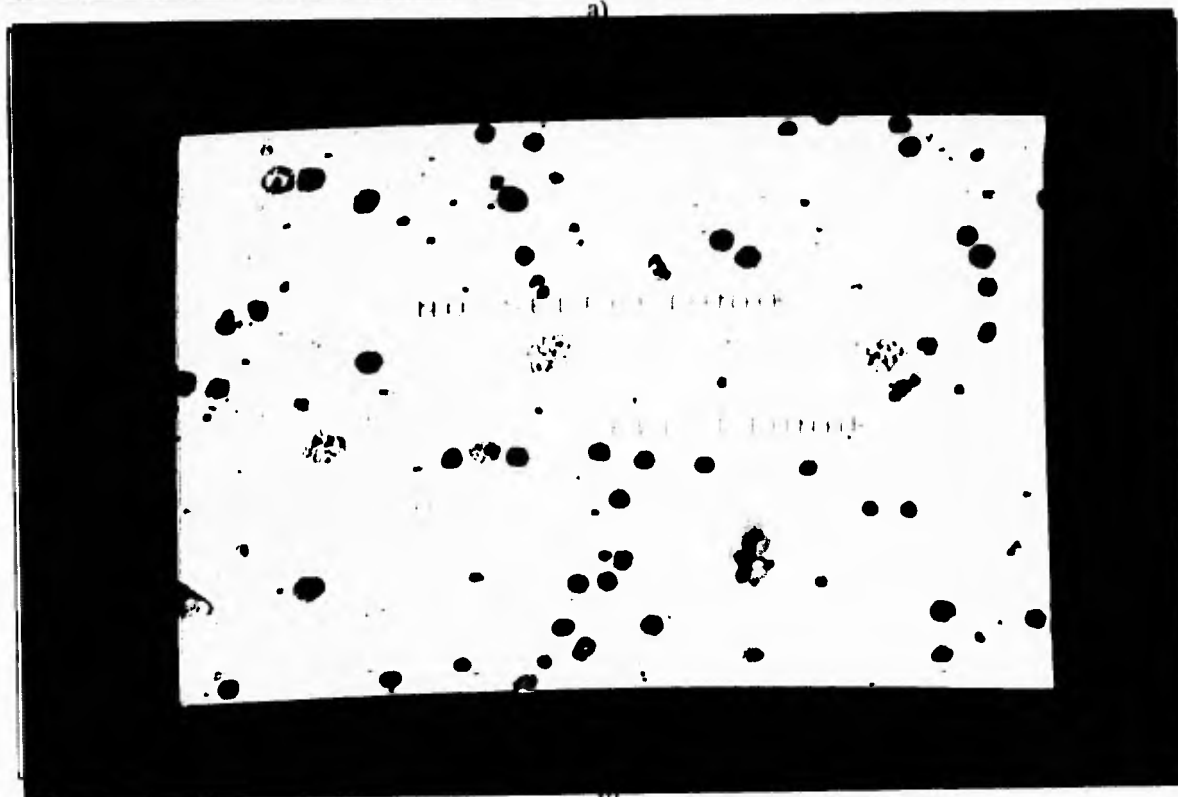
b)

Fig. A.E.3. Ejemplos de selección de mitosis cerradas en el entrenamiento. a) ejemplo 1. b) ejemplo 2.

Algunos ejemplos de mitosis que se consideran abiertas:



a)



b)

Fig. A.E.4. Ejemplos de mitosis abiertas, cerca de cerradas en el entrenamiento. a) ejemplo 1. b) ejemplo 2.

Mitosis abiertas

Antes de seleccionar un contorno como mitosis abierta se debe oprimir la tecla *A*.

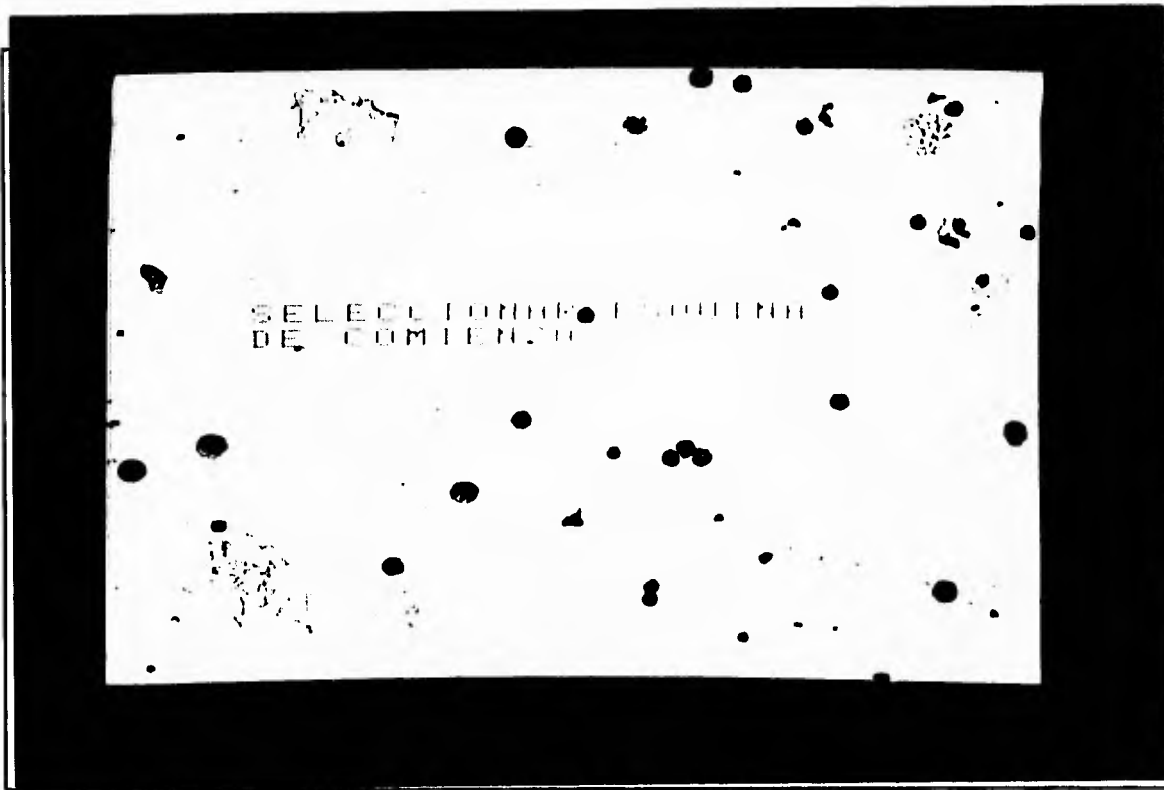
Cuando se ven una serie de cromosomas separados se deben seleccionar como mitosis abierta. Para hacer esto se encierran en una elipse.



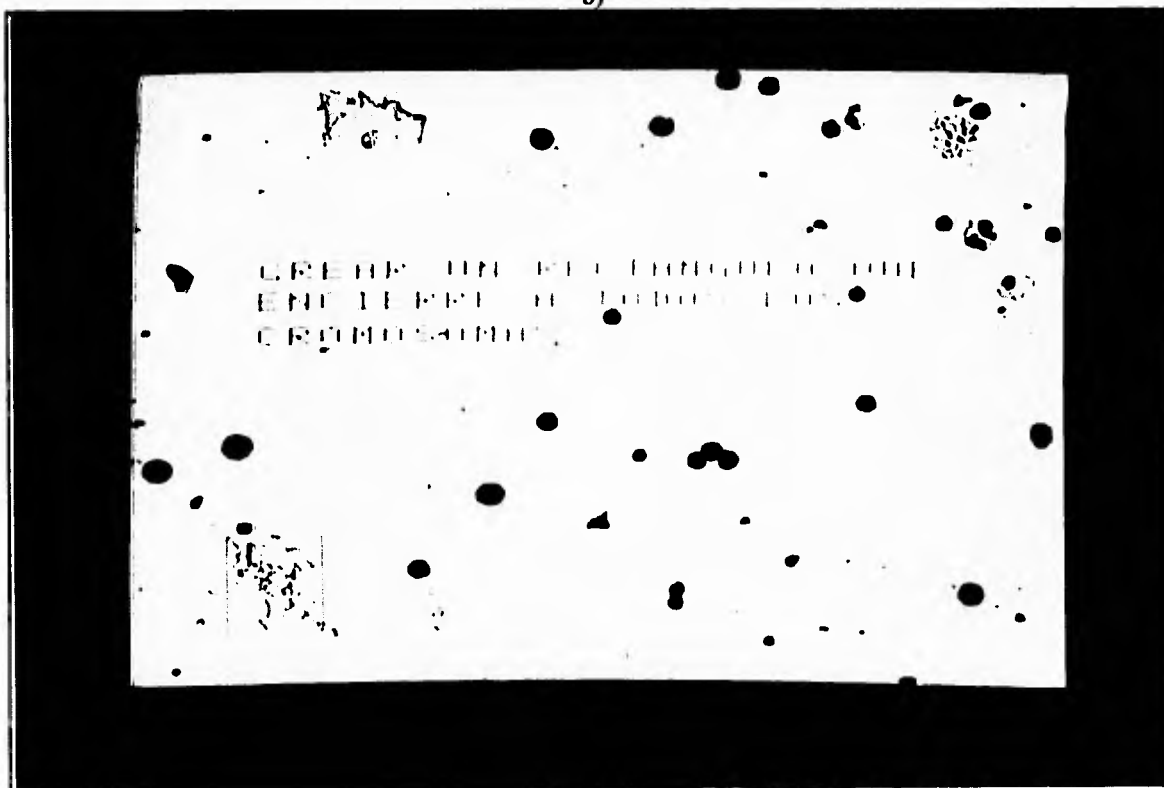
a)

Fig. A.E.5. Proceso de selección de mitosis abiertas.

a) primer paso. b) segundo paso. c) tercer paso. d) cuarto paso.



b)



c)

Fig. A.E.5. Proceso de selección de mitosis abiertas.
a) primer paso. b) segundo paso. c) tercer paso. d) cuarto paso.



d)

Fig. A.E.5. Proceso de selección de mitosis abiertas.
a) tercer paso. b) segundo paso. c) tercer paso. d) cuarto paso.

Se debe encerrar la mayor parte de la probable área de las mitosis cerradas.

También se debe evitar seleccionar los cuerpos que estén o puedan estar pegados a basura u otros cuerpos ya que si no se hace así se están dando datos falsos, es decir se está alterando un poco la muestra.



Fig A.E.6 Ejemplo de cuerpos que no deben ser seleccionados.

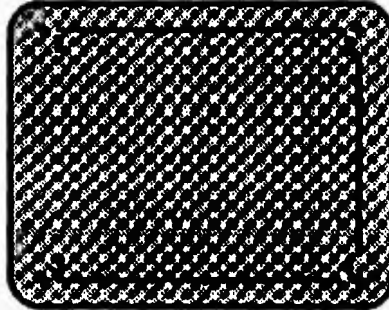
A continuación se dan las opciones con las que se cuentan al usar este comando:

- tecla A: selecciona mitosis abiertas.
- tecla B: deselectiona algún contorno.
- tecla C: selecciona mitosis cerradas.
- tecla E: selecciona núcleos estimulados.
- tecla Q: sale del comando para crear un entrenamiento nuevo.
- tecla Esc: sale del comando para crear un entrenamiento nuevo.
- botón derecho: termina la pantalla actual y realiza los cálculos de los objetos seleccionados.

DEFINICION DE OBJETIVOS

Para definir un nuevo objetivo, o para redefinir alguno, se debe llamar al comando **OBJETIVO DEFINIR**.

Se debe dar el valor del aumento del objetivo a definir y luego seleccionar un objeto de la pantalla que será movido a cuatro cruces que aparecerán en la pantalla gráfica:



Una vez que se coloca el objeto sobre alguna cruz se debe validar su llegada con el botón izquierdo del ratón.

Si la distancia C_3C_1 es diferente a C_4C_2 en un porcentaje mayor al 5% entonces se despliega un error de que las platinas están mal acomodadas. Lo mismo sucede con las distancias C_4C_3 y C_2C_1 . Esto se puede resolver si se acomodan bien las platinas o la cámara.

Normalmente al definir un objetivo se debe definir el traslape para dicho objetivo.

DEFINICION DE TRASLAPE

El traslape también se define de manera interactiva. Para definir la cantidad de traslape que cada objetivo posee se debe seleccionar un objeto, que tenga el tamaño típico de aquellos cuerpos que se desean sean contados, de la pantalla "gráfica" y colocarlo en alguna esquina de ella.

Una vez hecho esto el sistema pide que se haga desaparecer este objeto de la pantalla en el eje X. Con esto se define la cantidad de traslape en X. Lo mismo se hace para el eje Y.

Se pueden definir un traslape diferente al que el sistema tiene almacenado en el archivo OVERLAP.PAR. Si es así el sistema pregunta si se debe actualizar dicho archivo.

BARRIDOS

Para definir los barridos simplemente se pide que se le señalen al sistema las esquinas superior izquierda e inferior derecha de la zona de barrido.



El sistema guarda estas posiciones para cada laminilla que haya sido instalada.

Cuando el barrido se quiere hacer individual para cada laminilla entonces el sistema pide las esquinas superior izquierda e inferior derecha de cada laminilla instalada. El traslado a cada laminilla será hecho por el sistema automáticamente.

Es importante recordar que antes de definir el barrido es necesario introducir al sistema el número de laminillas que serán analizadas de otra manera solo definirá el barrido para aquel número que este guardado en memoria.

VERIFICACION

Al llamar este comando se debe seleccionar algún archivo con terminación REC que se halle en el directorio de trabajo. Esta selección se lleva a cabo de forma interactiva, si no se selecciona ningún archivo se tratará de abrir el archivo NUCLEOS.REC.

Cuando se revisa algún archivo de terminación REC se pueden usar las siguientes opciones:

botón derecho o tecla A: avanza a la pantalla siguiente.

tecla B: borra el plano gráfico de la pantalla "gráfica".

tecla C: hace que la verificación busque los campos donde hubo núcleos estimulados.

tecla D: busca las pantallas que contengan un cierto tipo de objetos.

tecla E: ajusta el enfoque en el campo actual.

tecla F: hace que la verificación busque los campos donde hubo cromosomas cerrados.

tecla G: recupera los símbolos de la pantalla "gráfica".

tecla H: hace que la verificación busque los campos donde hubo cromosomas abiertas.

tecla I: busca siguiente portaobjeto.

tecla M: se mueve n pantallas hacia adelante.

tecla N: hace que la verificación busque los campos contiguos.

tecla P: inserta un objeto nuevo en el archivo.

botón I o tecla R: regresa a la pantalla anterior.

tecla U: cambia los objetos mal clasificados en un campo de análisis.

tecla W: quita un objeto del archivo. Actualiza los resultados.

tecla Q o tecla Esc: sale de la rutina de verificación.

Apéndice B

Archivo de Soporte

FUENTES

**SRC0=INIMITC
SRC1=MITBASC0
SRC2=MITBASC1
SRC3=MITBASC2
SRC4=MITBASC3
SRC5=MITBASC4
SRC6=MITBASC5
SRC7=MITBASC6
SRC8=MITBASC7
SRC9=MITONLYC
SRC11=MITSUBC1
SRC12=MITSUBC2
SRC13=MITSUBC3
SRC14=MITSUBC4
SRC15=MITSUBC5
SRC16=MITSUBC6
SRC17=MITSUBC7
SRC18=MITSUBC8
SRC19=MITSUBC9
SRC20=MC2000
SRC21=mitosis
SRC22=mitcom
SRC23=mitcom2
SRC24=mitsub
SRC25=mitsub1
SRC26=leccom**

OBJETOS

**OBJ0=INIMITC.OBJ
OBJ1=MITBASC0.OBJ
OBJ2=MITBASC1.OBJ
OBJ3=MITBASC2.OBJ
OBJ4=MITBASC3.OBJ
OBJ5=MITBASC4.OBJ
OBJ6=MITBASC5.OBJ
OBJ7=MITBASC6.OBJ
OBJ8=MITBASC7.OBJ
OBJ9=MITONLYC.OBJ
OBJ11=MITSUBC1.OBJ
OBJ12=MITSUBC2.OBJ**

OBJ13=MITSUBC3.OBJ
OBJ14=MITSUBC4.OBJ
OBJ15=MITSUBC5.OBJ
OBJ16=MITSUBC6.OBJ
OBJ17=MITSUBC7.OBJ
OBJ18=MITSUBC8.OBJ
OBJ19=MITSUBC9.OBJ
OBJ20=MC2000.OBJ
OBJ21=MITOSIS.OBJ
OBJ22=MITCOM.OBJ
OBJ23=MITCOM2.OBJ
OBJ24=MITSUB.OBJ
OBJ25=MITSUB1.OBJ
OBJ26=LECCOM.OBJ

ENCABEZADOS

INC=VIDCONT.H FUNC.H FUNCEXT.H INTVASCO.H TIPOS.H MC2000.H INTVASCO.H
BIOAUX.H
DEF=DEF.H
XDEF=EXTDEF.H
FORINC=FORMIT.H

PROGRAMAS

EJE=mitosis.exe
CC= cl
FF= fl
TP= type

BANDERAS

FLAGSC= /c /AL /G2 /FPi /Ot
FLAGSL= /ST:10000 /SE:512 /E /NOE /OL
FLAGSF= /c /G2 /Os /4Yd /FPi /AL

inicio de dependencias

\$(EJE): \$(OBJ0) \$(OBJ1) \$(OBJ2) \$(OBJ3) \$(OBJ4) \$(OBJ5) \$(OBJ6) \$(OBJ7)\
\$(OBJ8) \$(OBJ9) \$(OBJ11) \$(OBJ12) \$(OBJ13) \$(OBJ14) \$(OBJ15)\
\$(OBJ16) \$(OBJ17) \$(OBJ18) \$(OBJ19) \$(OBJ20) \$(OBJ21) \$(OBJ22)\
\$(OBJ23) \$(OBJ24) \$(OBJ25) \$(OBJ26)

link \$(FLAGSL)
 @C:\EDGAR\MITOBJ.BAT,MITOSIS,MITOSIS,@C:\EDGAR\HSLIB.BAT;
 # DIR *.TMP
 \$(OBJ0): \$(SRC0).C \$(INC) \$(DEF)
 \$(CC) \$(FLAGSC) \$(SRC0).C>\$(SRC0).TMP
 \$(OBJ1): \$(SRC1).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC1).C>\$(SRC1).TMP
 \$(OBJ2): \$(SRC2).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC2).C>\$(SRC2).TMP
 \$(OBJ3): \$(SRC3).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC3).C>\$(SRC3).TMP
 \$(OBJ4): \$(SRC4).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC4).C>\$(SRC4).TMP
 \$(OBJ5): \$(SRC5).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC5).C>\$(SRC5).TMP
 \$(OBJ6): \$(SRC6).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC6).C>\$(SRC6).TMP
 \$(OBJ7): \$(SRC7).C \$(INC) VERIFY.H \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC7).C>\$(SRC7).TMP
 \$(OBJ8): \$(SRC8).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC8).C>\$(SRC8).TMP
 \$(OBJ9): \$(SRC9).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC9).C>\$(SRC9).TMP
 \$(OBJ11): \$(SRC11).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC11).C>\$(SRC11).TMP
 \$(OBJ12): \$(SRC12).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC12).C>\$(SRC12).TMP
 \$(OBJ13): \$(SRC13).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC13).C>\$(SRC13).TMP
 \$(OBJ14): \$(SRC14).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC14).C>\$(SRC14).TMP
 \$(OBJ15): \$(SRC15).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC15).C>\$(SRC15).TMP
 \$(OBJ16): \$(SRC16).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC16).C>\$(SRC16).TMP
 \$(OBJ17): \$(SRC17).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC17).C>\$(SRC17).TMP
 \$(OBJ18): \$(SRC18).C \$(INC) VERIFY.H \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC18).C>\$(SRC18).TMP
 \$(OBJ19): \$(SRC19).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC19).C>\$(SRC19).TMP
 \$(OBJ20): \$(SRC20).C \$(INC) \$(XDEF)
 \$(CC) \$(FLAGSC) \$(SRC20).C>\$(SRC20).TMP

FORTRAN

**\$(OBJ21): \$(SRC21).FOR \$(FORINC)
\$(FF) \$(FLAGSF) \$(SRC21).FOR>\$(SRC21).TMP**
**\$(OBJ22): \$(SRC22).FOR \$(FORINC)
\$(FF) \$(FLAGSF) \$(SRC22).FOR>\$(SRC22).TMP**
**\$(OBJ23): \$(SRC23).FOR \$(FORINC)
\$(FF) \$(FLAGSF) \$(SRC23).FOR>\$(SRC23).TMP**
**\$(OBJ24): \$(SRC24).FOR \$(FORINC)
\$(FF) \$(FLAGSF) \$(SRC24).FOR>\$(SRC24).TMP**
**\$(OBJ25): \$(SRC25).FOR \$(FORINC)
\$(FF) \$(FLAGSF) \$(SRC25).FOR>\$(SRC25).TMP**
**\$(OBJ26): \$(SRC26).FOR \$(FORINC)
\$(FF) \$(FLAGSF) \$(SRC26).FOR>\$(SRC26).TMP**

Apéndice C

Rutinas del Sistema

Rutinas:

Nombre	Proposito	Regreso	Argumentos	Tipo	Lenguaje	Archivo
AFOCA	Rutina para selección de rutinas de enfoque del microscopio	NADA	VACIO	Dependiente de BIOCOM. Usa DECCOM y ERREUR	FORTRAN	MITCOM2.FOR
ATAFOR	Renueva la posición del ratón en XSOUM,YSOUM(BIOCOM)	NADA	VACIO	Dependiente de BIOCOM. Usa MOUS y CLIC	FORTRAN	MITCOM2.FOR
BARDEF	Rutina para selección de rutinas de barrido a definir de las platinas	NADA	VACIO	Dependiente de BIOCOM. Usa DECCOM y ERREUR	FORTRAN	MITCOM2.FOR
BARRIDO	Rutina para selección de rutinas de barrido de las platinas	NADA	VACIO	Dependiente de BIOCOM	FORTRAN	MITCOM2.FOR
CONSYS	Configuration Autoradio(BIOCOM)	NADA	VACIO	INDEPENDIENTE	FORTRAN	MITOSIS.FOR
CONTRA	Encuentra el contraste y el numero de cambios dentro de un contorno a partir del valor de umbral VAL. POLNUM contorno VAL división entre clases de gris (OTSU) CPH cambios por pixel horizontales debajo del nivel de gris VAL. CPV cambios por pixel verticales debajo del nivel de gris VAL. CH cambios por linea horizontales debajo del nivel de gris VAL. CV cambios por linea verticales debajo del nivel de gris VAL. CON contraste	NADA	INTEGER*2 POLNUM REAL*4 VAL,CPH,CPV REAL*4 CH,CV,CON	Debe de existir el contorno. Depende de la rutina OTSU. Usa varias funciones y variables de BIOCOM y MATROX (LIGDEB, ROWR)	FORTRAN	MITSUB.FOR
CURSOR	Auxilia al movimiento de cursor	NADA	INTEGER*2 ACT	Dependiente de BIOCOM. Usa MOUS	FORTRAN	MITCOM2.FOR
DESCON	Revisa si algún contorno fue seleccionado CONSEL por el ratón en la posición XSEL,YSEL	NADA	INTEGER*2 CONSEL INTEGER*2 XSEL INTEGER*2 YSEL	Dependiente de BIOCOM. Usa MACTRF y otras variables	FORTRAN	MITCOM2.FOR
EJECUTA	Rutina para selección de rutinas de ejecución de rutinas del índice mitótico	NADA	VACIO	Dependiente de BIOCOM. Usa DECCOM y ERREUR	FORTRAN	MITCOM2.FOR

EROSI	Erosiona los contornos reduciéndolos ciertas unidades. Afecta las tablas de contornos. CONSEL contorno MARGEN reducción del contorno XSEL,YSEL centro del contorno	NADA	INTEGER*2 CONSEL REAL*4 MARGEN INTEGER*2 XSEL INTEGER*2 YSEL	Debe existir el contorno CONSEL. Depende del centro de gravedad. Usa variables de BIOCOM, modifica a XPOINT y YPOINT.	FORTRAN	MITSUB.FOR
FMODA	Calcula la moda en gris de la pantalla. VAL valor de la moda de gris	NADA	INTEGER*2 VAL	Dependiente de BIOCOM y MATROX. Usa FINACQ e IHISEN	FORTRAN	MITSUB.FOR
FUNCIO	Función que aplica una función del tipo aX^4 o aX^2 a las tablas (LUT's). FAC valor de a TIP <4 => cuadrática TIP = 4 => cuarta	NADA	REAL*4 FAC INTEGER*2 TIP	Dependiente de BIOCOM y MATROX. Usa POLYNO, TABRES y otras variables.	FORTRAN	MITSUB.FOR
HISZON	A partir del cálculo del histograma de la imagen calcula la diferencia de pendientes entre los segmentos de recta C0-C1 y C1-C2 CONT contorno COMP valor de referencia para el porcentaje ocupado por los grises claros C0 punto inicial de la recta C0-C1 C1 punto final de la recta C0-C1 e inicial de la recta C1-C2 C2 punto final de la recta C1-C2 X porcentaje ocupado por los grises claros P1 pendiente de la recta C0-C1 P2 pendiente de la recta C1-C2	NADA	INTEGER*2 CONT,COMP INTEGER*2 C0,C1,C2,X REAL*4 P1,P2	Dependiente de BIOCOM y MATROX. Usa ZONZON, MGRAPH, LIGDEB, ROWR, LIGFIN y otras variables	FORTRAN	MITSUBI.FOR
INMORF	Inicia las tablas de Morfometría de BIOCOM	NADA	VACIO	Dependiente de BIOCOM. Usa INICON y otras variables	FORTRAN	MITCOM2.FOR
MITCOM	Rutina para selección de rutinas de MITOSIS	NADA	VACIO	Dependiente del PROG de BIOCOM	FORTRAN	MITCOM.FOR
OBJETIVOS	Rutina para selección de rutinas de objetivos del microscopio	NADA	VACIO	Dependiente de BIOCOM	FORTRAN	MITCOM.FOR

OTSU	Realiza proceso para encontrar un valor de umbral entre dos clases en un contorno. POLNUM contorno VAL valor que divide los niveles de gris del contorno	NADA	INTEGER*2 POLNUM REAL*4 VAL	Dependiente de BIOCOM y MATROX. Usa BALTRA, INIBAL y otras variables. Debe existir el POLNUM	FORTRAN	MITSUB.FOR
REGRES	Funcion que regresa las LUT's a su estado original	NADA	VACIO	Dependiente de BIOCOM y MATROX. Usa LUTENT, LUTACQ y otras variables	FORTRAN	MITSUB.FOR
TRAIN	Rutina para seleccion de rutinas de entrenamiento del sistema	NADA	VACIO	Dependiente de BIOCOM. Usa DECCOM y ERREUR	FORTRAN	MITCOM2.FOR
accion_mito	funcion que realiza la operacion de mitosis. <i>mito</i> apuntador al archivo de verificacion, normalmente NUCLEOS.REC <i>dato</i> valores de nucleos y mitosis hallados en la pantalla. Estructura. <i>error</i> < 0 si hubo error en la busqueda = 0 no hubo error en la busqueda	char 0 sin error 1 error	FILE *mito dato6 *dato char *error	Modifica al archivo NUCLEOS.REC pasado como *mito, modifica *dato y crea un archivo temporal.	C	MITSUBC3.C
accion_otro	funcion que realiza las funciones especificadas en un archivo cmd. Llena el buffer PHYSBUF de BIOCOM con nombre del archivo .CMD	NADA	char nombre[]	Modifica la variable PHYSBUF y POINTPHYS	C	MITSUBC4.C
actobj	activa objetivos. Interfaz con FORTRAN	NADA	VACIO	Necesita OBJETIV.MIC Modifica archivo OVERLAP.PAR y a las variables globales: overlap_existe, objetivo_definido, lim_inf_g, lim_inf_f y facry	C	MITBASC0.C
act_porta	actualiza los valores del portaobjetos	char 0 sin error 1 error	VACIO	actualiza los valores de la laminilla en revision por la rutina verify	C	MITSUBC8.C

acttr	funcion que sirve para activar un archivo de entrenamiento. Interfaz con FORTRAN	NADA	VACIO	Modifica ENTRENA.PAR, las variables entrena_existe, entrena_activo, estimul, cromo_ab y cromo_cer. Usa funcion inicia limites	C	MITBASC4.C
actual_rata	actualizacion del movimiento del raton po:rx indica la posicion de inicio y final en x del raton po:ry indica la posicion de inicio y final en y del raton.	char	int *poxx,*poxy	Modifica XSOURI,YSOURI Usa CATEG y SOUPOS	C	MITSUBC7.C
adqjoy	funcion que activa adquisicion continua al mover joystick. Interfaz con FORTRAN	NADA	VACIO	Modifica el tipo de adquisicion. Funcion que hace una revision continua del controlador MC2000	C	MITSUBC4.C
advertis	Solamente letrero para la rutina verify x0,y0 indican la esquina superior derecha del rectangulo de la pantalla donde se pondra el letrero x1,y1 indican la esquina superior derecha del rectangulo de la pantalla donde se pondra el letrero	NADA	int x0,y0,x1,y1	Pinta un letrero de las opciones de la funcion de verificacion del archivo NUCLEOS.REC, verify.	C	MITSUBC8.C
ajusta_color	valida el color que usa BIOCUM para el fondo de pantalla	int regresa el valor que BIOCUM usa como fondo.	int color Valor de la estructura INDIGO[0][3] de BIOCUM	Depende de la estructura INDIGO de BIOCUM.	C	MITSUBC6.C
arch_update	actualiza las modificaciones en el archivo de verificacion archivo (NUCLEOS.REC).	char 0 sin error 1 error	FILE *archivo	Modifica el archivo de verificacion (NUCLEOS.REC). Actualiza las modificaciones hechas al archivo de verificacion.	C	MITSUBC8.C

asocia	funcion que asocia portaobjetos de control con sus experimentos	NADA	VACIO	Modifica las variables globales asociados[] y controles[] usando num portaobj	C	MITSUBC4.C
aux_datos	funcion para ver características	NADA	int option		C	MITONLYC.C
avante	rutina de avance tipo por: 0..... pantallas 1..... mitosis abiertas 2..... mitosis cerradas file es el archivo donde se mueve (NUCLEOS.REC) posact indica la posicion actual del apuntador de archivo. La actualiza al salir cambio es una variable usada para indicar si se debe cambiar de campo.	char 0 sin error 1 error	char tipo FILE *file long *posact unsigned char *cambio	Cambia el valor de posact al campo de file donde encuentra un objeto tipo. Modifica cambio y archivo (NUCLEOS.REC). Se usa dentro de la rutina verify exclusivamente.	C	MITSUBCR.C
barpre	funcion que tiene predefinido un barrido. Interfaz con FORTRAN	NADA	VACIO	Modifica la variable global: region[num portaobj]	C	MITBASC1.C
borra_rect	funcion que borra un rectangulo de la pantalla grafica ima indica el plano imagen actual x0,y0 indican la esquina superior derecha del rectangulo x1,y1 indican la esquina inferior izquierda del rectangulo	NADA	int ima,x0,y0,x1,y1	Borra un rectangulo de la pantalla de imagen. Se usa al seleccionar una mitosis abierta, funcion elipse.	C	MITSUBC7.C
busqueda	funcion que revisa la existencia de algun objetivo dentro del archivo objetiv.mic. reg valor del objetivo arch apuntador a la estructura de objetiv.mic.	long int >=0 la posicion del objetivo dentro del archivo objetiv.mic. < 0 no esta definido.	int reg FILE *arch	Usada en defobj	C	MITSUBC2.C
cal	Manda la Instruccion de calibracion al controlador MC2000.	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C

calc_cont	<p>funcion que elije y calcula los parametros de los contornos en el entrenamiento</p> <p><i>cambio</i> indica si se realizo algun calculo de cuerpos dentro de la rutina calc_cont</p> <p><i>le, la</i> y <i>lc</i> indican el numero de nucleos estimulados, mitosis abiertas y cerradas al entrar a calc_cont.</p>	acum	<p>char *cambio</p> <p>int la,lc,ld</p>	<p>Permite elegir y calcular en una pantalla los valores $\Sigma x, \Sigma x^2$ y <i>n</i> de los parametros de los Nucleos, Mitosis Abiertas y Cerradas. Se usa dentro de <i>nuetr</i></p>	C	MITSUBC7.C
calc_desvymed	<p>funcion que calcula deriv estandar y media bajo la formula:</p> $\sqrt{\left(\left(\frac{sumax2}{n}\right) - \left(\frac{sumax}{n}\right)^2\right)}$ <p>donde <i>sumax</i> sumatoria de los elementos</p> <p><i>sumax2</i> sumatoria de los elementos²</p> <p><i>n</i> numero de elementos</p>	<p>regresa una estructura del tipo <i>dato2</i> que contiene la media y la desviacion estandar</p>	<p>float sumax,sumax2</p> <p>int n</p>	<p>Depende de <i>math.h</i></p>	C	MITSUBC5.C
calc_limite_enff	<p>funcion que calcula limite de movimiento para el enfoque fino. Necesita del tipo de objetivo a ser usado para calcular dicho limite</p>	<p>int</p> <p>limite de movimiento para el enfoque</p>	<p>int tipo</p>	<p>Funcion experimental sobre los limites para el enfoque grueso. Usarla fuera de este programa no tiene sentido.</p>	C	MITSUBC6.C
calc_limite_enfg	<p>funcion que calcula limite de movimiento para el enfoque grueso. Necesita del tipo de objetivo a ser usado para calcular dicho limite</p>	<p>int</p> <p>limite de movimiento para el enfoque</p>	<p>int tipo</p>	<p>Funcion experimental sobre los limites para el enfoque grueso. Usarla fuera de este programa no tiene sentido.</p>	C	MITSUBC6.C

calcula_mov	funcion que calcula el desplazamiento de los portaobjetos. <i>porciento</i> indica la sobreposicion de las tomas en x y en y <i>objeto</i> indica el tamaño de la pantalla en x e y para el objetivo definido <i>veces</i> guarda el numero de desplazamientos en x e y permitidos en cada laminilla <i>salida</i> indica la magnitud del desplazamiento en x e y.	char 0 sin error 1 error, no hay objetivo definido o la magnitud del desplazamiento es negativa.	dato% porciento coord objeto dato1 veces[] coord *salida	Modifica a veces[] y a salida	C	MITSUBC4.C
changes	Funcion para cambiar simbolos y cuenta en el archivo NUCLOES.REC <i>archivo</i> indica el apuntador del archivo de verificacion (NUCLEOS.REC). <i>posact</i> indica la posicion donde hara los cambios dentro del archivo <i>alter</i> se modifica a uno si hubo cambios	char 0 sin error 1 error	FILE *archivo long posact unsigned char *alter	Modifica alter y el archivo (NUCLEOS.REC). Se usa dentro de la rutina verify exclusivamente.	C	MITSUBC8.C
clreol	rutina que borra una linea y	NADA	int y	Depende del encabezado dos.h de TC o MC.	C	MITSUBC1.C
clrscr	rutina que borra la pantalla con un color.	NADA	int color	Usa variable global BUF1 e IMACOU. Usa MGRAPH, NMFACQ. Detiene por un instante la adquisicion continua mientras calcula	C	MITSUBC1.C
cls_par	rutina que borra una ventana con un color y de coordenadas (x1,y1)-(x2,y2)	NADA	int x1,y1,x2,y2,color	Depende del encabezado dos.h de TC o MC.	C	MITSUBC1.C
conteo	funcion que obtiene el numero de nucleos a contar. Interfaz con FORTRAN	NADA	VACIO	Modifica la variable global num_nucleos	C	MITBASC3.C
crea_rect	funcion que crea un rectangulo en la pantalla grafica <i>ima</i> indica el plano imagen actual <i>x0,y0</i> indican la esquina superior derecha del rectangulo <i>x1,y1</i> indican la esquina inferior izquierda del rectangulo	NADA	int ima,x0,y0,x1,y1	Plota un rectangulo en la pantalla de imagen segun el movimiento del raton. Se usa al seleccionar una mitosis abierta, funcion elipse.	C	MITSUBC7.C

crucos	funcion que pone cruces en el plano imagen	NADA	VACIO	Usada en defobj	C	MITSUBC2.C
cursori	funcion que pinta el cursor en la pantalla <i>n</i> veces	NADA	unsigned char <i>n</i>	Debe ser usada dentro de BIOCUM, depende de la rutina CURS.	C	MITSUBC5.C
deflx8	funcion que define el barrido de un solo porta objetos para las 8 laminillas. Interfaz con FORTRAN	NADA	VACIO	Modifica la variable global: region[num portaobj]	C	MITBASC1.C
defind	funcion que define un barrido independiente para cada porta objetos. Interfaz con FORTRAN	NADA	VACIO	Modifica a la variable global: region[laminilla]	C	MITBASC1.C
defobj	define objetivos. Interfaz con FORTRAN	NADA	VACIO	Modifica al archivo OBJETIV.MIC	C	MITBASC0.C
despx	funcion que define desplazamiento en <i>x</i> y maximo movimiento en <i>x</i> . Interfaz con FORTRAN	NADA	VACIO	Modifica DESP.PAR y variables maximo_y, objetivo_existe, objetivo_definido y offset_lam	C	MITBASC3.C
dim3	Activa las mediciones en tres dimensiones en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
elipse	funcion que valida el dibujo de un circulo que encierra una mitosis <i>tipo</i> indica un contorno que se desea este dentro de la elipse, si el contorno no esta dentro de la elipse la elipse se no se crea. Un valor 0 evita esta opcion. <i>n</i> el numero de mitosis abierta que se esta encerrando en la elipse. <i>posx, posy</i> posicion actual del raton <i>rx, ry</i> radios en <i>x</i> e <i>y</i> de la elipse <i>h, k</i> indican el centro de la elipse.	char 0 sin error 1 error	char tipo int <i>n</i> short contorno int *posx, *posy float *rx, *ry int *h, *k	Valida que la elipse no sea un solo punto.	C	MITSUBC7.C
ellipse	funcion que pinta una elipse por medio de la funcion vec <i>h, k</i> indican el centro de la elipse <i>radx, rady</i> indican los radios en <i>x</i> e <i>y</i>	NADA	float <i>h, k, radx, rady</i>	Pinta una elipse en el plano actual usando la funcion VEC	C	MITSUBC7.C

encima	funcion que define el traslape de las tomas de video. Interfaz con FORTRAN	NADA	VACIO	Modifica archivo OVERLAP.PAR y variables overlap_existe y overlap	C	MITBASC7.C
enfoque	enfoca el microscopio tipo = 1 fino tipo = 0 grueso	char 0 sin error 1 error al enfocar	char tipo	Modifica off_z y off_z1	C	MITSUBC5.C
ensena	funcion que solo enseña la posicion de las platinas. Interfaz con FORTRAN	NADA	VACIO	Usa la variable global overlap	C	MITBASC7.C
entero	funcion que regresa un caracter dentro de los numeros digitales desde el teclado	int funcion que regresa un caracter dentro de los numeros digitales desde el teclado	VACIO	Universal Usa getch().	C	MITSUBC4.C
espera_resp	funcion que espera una tecla para iniciar una nueva pantalla para un entrenamiento nuevo. y linea donde se pintara el mensaje de espera	char regresa la tecla oprimida	int y	Espera hasta que el usuario oprima la tecla Q, I o el boton Izq del raton (usa CLIC).	C	MITSUBC4.C
figura	funcion que pinta un porta objetos en la pantalla. i indica el numero del portaobjetos x,y la esquina desde dondes se dibuja tipo el tipo de laminilla: C control E experimento L libre num el numero de asociacion, a quien esta asociado	NADA	int i, x, y char tipo,num	Pinta la figura de un portaobjetos en la pantalla usando caracteres ASCII. Usa gotoxy	C	MITSUBC4.C
findnext	funcion que busca la siguiente laminilla a analizar usando la rutina loccmd	int < 0 no hay otra laminilla. >=0 siguiente laminilla	VACIO	Modifica la variable controles[] cuando posee valores entre 1 y 8 pasa a 0	C	MITSUBC2.C

findnextcontrol	funcion que busca el siguiente control a revisar	int < 0 no hay otra laminilla control. >=0 siguiente laminilla control	VACIO	Modifica la variable controles[] cuando posee valor 67 pasa a 99	C	MITSUBC2.C
findnextexpasoc	funcion que busca el siguiente experimento asociado a control	int < 0 no hay otra laminilla experimento asociada al control. >=0 siguiente laminilla asociada al control	int control	Modifica la variables globales asociados[] y controles[]	C	MITSUBC2.C
findnextlibre	funcion que busca el siguiente portaobjetos libre	int < 0 no hay otra laminilla libre. >=0 siguiente laminilla libre	VACIO	Modifica la variable global controles cuando posee valor 76 pasa 101	C	MITSUBC2.C
find_umbral	funcion que encuentra el umbral de una imagen	int regresa el valor de umbral optimo por metodo de OTSU de dos clases	VACIO	Usada dentro de seg_new. Modifica HISBUF	C	MITSUBC5.C
fino	enfoco fino. Interfaz con FORTRAN	NADA	VACIO	Modifica las variables globales off z y off z1	C	MITBASC3.C
gotoxy	rutina para colocar el cursor en alguna posicion x,y de la pantalla	NADA	int x, y	Usa encabezado dos.h	C	MITSUBC1.C
graf_sel	Rutina para seleccionar un archivo de entrenamiento REC para la verificacion. Interfaz con el raton. *name sera el nombre del archivo seleccionado *exito brinda informacion sobre si hubo seleccion o no.	char 0 sin error 1 error	char *name char *exito	Mut parecida a la rutina actr	C	MITSUBC9.C
grueso	enfoco grueso. Interfaz con FORTRAN	NADA	VACIO	Modifica las variables globales off z y off z1	C	MITBASC3.C

habil_joy	Habilita el uso del joystick en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
habil_x	Habilita el movimiento en X en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
habil_y	Habilita el movimiento en Y en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
habil_z	Habilita el movimiento en Z en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
imp_porta	imprime el valor del portaobjetos	char 0 sin error 1 error	VACIO	Imprime los valores del portaobjetos en revision actual por la rutina verify	C	MITSUBC8.C
imprime_train	funcion que imprime los valores del entrenamiento actual	NADA	VACIO	Usada en aux_datos	C	MITSUBC2.C
inactiva_xy	Inhibe el movimiento en X e Y en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
incuer	informa el numero de cuerpos a contar. Interfaz con FORTRAN	NADA	VACIO	Usa la variable num_nucleos	C	MITBASC3.C
in_elips	funcion que valida si un punto esta dentro los limites de una elipse h,k indican el centro de la elipse rx,ry radios en x e y de la elipse x,y coordenadas del punto a validar	char 0 sin error 1 error	float h,k,rx,ry,x,y	Verifica si un punto x,y esta dentro los limites de una elipse con parametros x,y,rx,ry,h,k	C	MITSUBC7.C
infr	informa el entrenamiento que esta activo. Interfaz con FORTRAN	NADA	VACIO	Usa variables de BIOCOM y entren activo	C	MITBASC3.C
inhib_joy	Inhibe el uso del joystick en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
inhib_x	Inhibe el movimiento en X en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C

inhib_y	Inhibe el movimiento en Y en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
inhib_z	Inhibe el movimiento en Z en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
inhiz	Inhibe el eje Z en el controlador MC2000 para las rutinas de medicion y calibracion.	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
inicia	Interfaz para inicializar las platinas. Interfaz con FORTRAN	NADA	VACIO	Requiere de las rutinas del controlador MC2000	C	INIMITC.C
inicia_limites	funcion que inicia las variables globales est, cr_ab y cr_cer usadas en el reconocimiento del indice mitotico.	NADA	VACIO	Depende de la existencia de las variables globales est, cr_ab y cr_cer usadas en el reconocimiento del indice mitotico.	C	MITSUBC5.C
ini_cont1	Formula para la obtencion del contraste en la pantalla. <i>inicio</i> es la posicion de la primera linea a muestrear. <i>incr</i> indica las siguientes lineas al muestrear en forma de incremento desde inicio. <i>plano</i> hace referencia al plano de la MATROX a muestrear.	int regresa el valor del contraste en dicha pantalla.	int inicio,incr,plano	Usa encabezado dos.h	C	MITSUBC1.C
ini_puerto	inicializa el puerto serie COM2 con 4800,n,8,1	char 0 sin error 1 error	VACIO	Requiere de dos.h y del puerto serial COM1	C	MC2000.C
ins_arch	inserta nodos, usados para la impresion de un menu (en pantalla) de archivos disponibles, en archivo temporal. <i>nombre</i> nombre del archivo disponible <i>arch</i> apuntador al archivo temporal. <i>posx</i> la posicion X del letrero nombre sobre la pantalla. <i>posy</i> la posicion Y del letrero nombre sobre la pantalla.	char 0 sin error 1 error, problemas de escritura en el archivo arch. No pudo crearse el nuevo nodo.	char nombre[] FILE *arch unsigned char *posx,*posy	Es ad hoc para usarse dentro de la rutina actr. Sin embargo puede ser usada para la creacion de menus. Calcula las posiciones de los archivos *.ENT en la pantalla.	C	MITSUBC5.C

ins_del	funcion para insertar un nodo en una estructura ligada del tipo nod_redo con apuntadores idel, adel y del pos indica la posicion que sera borrada del archivo	char 0 sin error 1 error	long int pos	Lista que contiene las posiciones que seran borradas del archivo de verificacion. Modifica idel, adel y del. Uso exclusivo en rutina verify	C	MITSUBC8.C
inserta	inserta nodos en una lista lineal doblemente ligada usada para la impresion de un menu, en pantalla, de archivos disponibles en el directorio actual. nombre nombre del archivo disponible ap apuntador movil sobre la lista ini apuntador que marca el inicio de la lista fin apuntador que marca el fin de la lista posx la posicion X del letrero nombre sobre la pantalla. posy la posicion Y del letrero nombre sobre la pantalla.	char 0 sin error 1 error, ya no hay memoria disponible. No pudo crearse el nuevo nodo.	char nombre[] node **ap, **ini, **fin unsigned char *posx, *posy	Es ad hoc para usarse dentro de la rutina actr. Sin embargo puede ser usada para la creacion de menus. Calcula las posiciones de los archivos *.ENT en la pantalla.	C	MITSUBC5.C
ins_ins	funcion para insertar un nodo en una estructura ligada del tipo nod_redo con apuntadores iins, ains y ins e numero de nucleos estimulados c numero de mitosis cerradas a numero de mitosis abiertas	char 0 sin error 1 error	float e,c,a	Lista que contiene los datos a ser insertados dentro del archivo de verificacion. Modifica iins, ains e ins. Uso exclusivo en rutina verify	C	MITSUBC8.C
ins_pila	inserta nodos en una pila ligada ap es el apuntador movil de la pila ligada fin apuntador que marca el fin de la pila ligada x0,y0 indican el centro de la elipse rx,ry indican el radio en x e y num indica en numero de elipse que se guardo	char 0 sin error 1 error	nod_pila **ap, **fin int x0,y0,rx,ry unsigned char num	Guarda los parametros de una elipse en cada nodo. Se usa al seleccionar una mitosis abierta.	C	MITSUBC7.C

ins_redo	funcion para insertar un nodo en una estructura ligada del tipo nod_redo con apuntadores iredul, aredul y resul e numero de nucleos estimulados c numero de mitosis cerradas a numero de mitosis abiertas	char 0 sin error 1 error	char n float e,c,a	Lista que contiene los datos de cuerpos hallados en cada laminilla. Usada en lee_porta. Modifica iredul, aredul y resul. Se usa exclusivamente en rutina verify	C	MITSUBC8.C
leecmd	funcion que ejecuta el barrido y ejecuta comandos de un archivo cmd. Interfaz con FORTRAN	NADA	VACIO	Modifica las variables globales BAN1 y LOGBUF	C	MITBASC2.C
lee_num	funcion que valida entrada de numeros inicio da la posicion inicial para comenzar a leer v tam el tamaño de la cadena	char * Regresa una cadena con solo numeros	unsigned inicio char cadena[] unsigned tam	C estandar Independiente	C	MC2000.C
lee_porta	lee portaobjetos antes de empezar la comprobacion del archivo de verificacion (NUCLEOS.REC)	char 0 sin error 1 error	VACIO	Modifica a iredul, aredul y resul. Uso exclusivo en rutina verify	C	MITSUBC8.C
letrero_nuotr	pinta el letrero de un nuevo entrenamiento. le numero de nucleos la numero de mitosis abiertas lc numero de mitosis cerradas	NADA	int le,la,lc	Usada en calc_cont y en nuotr	C	MITSUBC2.C
limpia_pantalla	limpia pantalla para el nuevo entrenamiento actual indica el contorno seleccionado categ la nueva categoria para dicho contorno ap se usa para cambiar categorias de mitosis abiertas fin se usa para cambiar categorias de mitosis abiertas	NADA	short actual int categ nod_pila **ap,** fin	Modifica CATEG[]. La estructura ligada nod_pila controla las elipses que encierran a las mitosis abiertas	C	MITSUBC6.C
mayus	hace la conversion a mayusculas	char Si car es una letra entonces la regresa como mayuscula. De otra manera regresa car.	char car	Requiere de la funcion getch() para DOS	C	MC2000.C

mide_gris	funcion para obtener el nivel de gris y su desviacion estandar en una area circular de la pantalla. <i>x,y</i> centro del circulo. <i>rx,ry</i> radios mayor y menor de la elipse. <i>h,k</i> desplazamiento del centro de la elipse con respecto al origen del plano. <i>dev</i> posee el valor de la desviacion estandar del nivel de gris.	float regresa la media de gris de una region eliptica.	int x,y,rx,ry,h,k float *dev	Una funciones de la MATROX directamente. fg_pur	C	MITSUBC2.C
mit	funcion que realiza el barrido y obtiene el indice mitotico. Interfaz con FORTRAN	NADA	VACIO	Modifica al archivo NUCLEOS.REC, necesita las variables globales <i>entrena_existe</i> , <i>overlap_existe</i> , <i>region[]</i> , <i>num_portaobj</i> , <i>asociados[]</i> , <i>controles[]</i> .	C	MITBASC2.C
mitcmd	funcion que ejecuta el barrido, ejecuta comandos de un archivo cmd y obtiene el indice mitotico. Interfaz con FORTRAN	NADA	VACIO	Modifica las variables globales AC, BAN2 y LOGBUF. Modifica el archivo NUCLEOS.REC y necesita las variables globales <i>entrena_existe</i> , <i>overlap_existe</i> , <i>region[]</i> , <i>num_portaobj</i> , <i>asociados[]</i> , <i>controles[]</i> .	C	MITBASC2.C
modo_host	Pone en modo host al controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C

mueve_pobj	<p>funcion que mueve los portaobjetos para uso de CMD's segun desplazamiento.</p> <p><i>i</i> representa el numero de laminilla en la que se mueve.</p> <p><i>j</i> el paso en el eje y de las platinas</p> <p><i>k</i> el paso en el eje x de las platinas</p> <p><i>factorx</i> representa el sentido y la magnitud del desplazamiento en x</p> <p><i>factory</i> representa el sentido y la magnitud del desplazamiento en y</p> <p><i>orig</i> indica si es el primer campo</p> <p>0 no lo es=> realiza enfoque</p> <p>1 si lo es => no realiza enfoque</p> <p><i>mit</i> indica si es proceso de indice mitotico</p> <p>= 1 habilita enfoque junto con <i>afo</i></p> <p><i>afo</i> indica que debe enfocar fino cada vez que se mueve. Al cambiar de laminilla enfoca grueso.</p>	<p>char</p> <p>0 se movio sin error</p> <p>1 hubo error al moverse</p>	<p>int <i>ban,*i,*j,*k</i></p> <p>float <i>*factorx</i></p> <p>float <i>*factory,orig</i></p> <p>int <i>mit</i></p> <p>char <i>afo</i></p>	<p>Usa variables <i>region[]</i> y <i>veces[]</i>.</p> <p>Modifica variable <i>off_z</i></p> <p>Modifica <i>ij,k_factorx</i> y <i>factory</i></p>	C	MITSUBC4.C
muevexyz	<p>funcion que manda comando de movimiento en x,y,z:</p> <p>tipo:</p> <p>0——absolutas</p> <p>1——relativas</p> <p>Espera a que terminen de moverse para salir</p>	<p>char</p> <p>0 sin error</p> <p>1 error</p>	<p>float x, y, z</p> <p>int tipo</p>	<p>Requiere del controlador MC2000 y del puerto serial COM1</p>	C	MC2000.C
nuetr	<p>crea un nuevo entrenamiento. Interfaz con FORTRAN</p>	NADA	VACIO	<p>Crea un nuevo archivo *.ENT. Modifica ENTRENA.PAR, las variables <i>entrena_existe</i>, <i>entrena_activo</i>, <i>estimul</i>, <i>chromo_ab</i> y <i>chromo_cer</i>.</p> <p>Usa funcion <i>inicia limites</i></p>	C	MITBASC5.C

numbioc	funcion para leer los numeros del menu de BIOCOM. num indica el mensaje a desplegar tipo =1 => real tipo =0 =>entero ni valor entero nf valor real	char 0 es numero 1 no es numero	int num char *tipo long *ni float *nf	Modifica la variable global LOGBUF	C	MITBASC7.C
numpor	funcion que obtiene el numero de portaobjetos a usar. Interfaz con FORTRAN	NADA	VACIO	Modifica la variable global num portaobj	C	MITBASC1.C
mover	funcion que mueve las platinas a una posicion x,y,z. Interfaz con FORTRAN	NADA	VACIO	Mueve las platinas a una posicion x,y,z.	C	MITBASC7.C
obt_pos	funcion que obtiene las coordenadas x,y y z segun el valor de la variable dim, la cual indica el numero de dimensiones a obtener	char 0 sin error 1 error	coord3 *res	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
only	funcion temporal para el reconocimiento de patrones de la pantalla actual. Interfaz con FORTRAN	NADA	VACIO	Usa encabezado dos.h	C	MITONLY.C
ordenl	Genera el orden de busqueda para las laminillas en la rutina leocmd	char 0 sin error 1 error	VACIO	Modifica a la variable global controles[]	C	MITSUBC4.C
oversh	funcion que enseña el valor de trasape de las tomas de video. Interfaz con FORTRAN	NADA	VACIO	Usa rutinas MESSA y a la variable oversh	C	MITBASC7.C
parte	funcion de afilado provisional. Interfaz con FORTRAN	NADA	VACIO	Modifica los la tabla de contornos, afina la imagen. Provisional.	C	MITBASC7.C
percen	funcion que obtiene el porcentaje minimo para cada control. Interfaz con FORTRAN	NADA	VACIO	Modifica variable global percent	C	MITBASC7.C
pone	funcion para insertar simbolos en la cuenta del archivo de verificacion (NUCLEOS.REC) archivo indica el apuntador del archivo de verificacion (NUCLEOS.REC). posact indica la posicion donde hara los cambios dentro del archivo alter se modifica a uno si hubo inserciones	char 0 sin error 1 error	FILE *archivo long posact unsigned char *alter	Modifica alter y el archivo (NUCLEOS.REC). Se usa dentro de la rutina verify exclusivamente.	C	MITSUBC8.C

por100	funcion que cambia el valor del porcentaje en y. Interfaz con FORTRAN	NADA	VACIO	Modifica archivo DESP.PAR y variables global porcy y minimo_y. Usa barpre.	C	MITBASC3.C
quita	funcion para quitar simbolos en la cuenta del archivo de verificacion (NUCLEOS.REC) <i>archivo</i> indica el apuntador del archivo de verificacion (NUCLEOS.REC). <i>posact</i> indica la posicion donde hara los cambios dentro del archivo <i>alter</i> se modifica a uno si hubo inserciones	char 0 sin error 1 error	FILE *archivo long posact unsigned char *alter	Modifica alter y el archivo (NUCLEOS.REC). Se usa dentro de la rutina verify exclusivamente.	C	MITSUBC3.C
recCromosomasAbiertos	funcion que reconoce los cromosomas abiertos. <i>arch</i> apuntador a archivo temporal que sirve para almacenar los nucleos hallados en una pantalla. <i>dato</i> contiene el numero de mitosis abiertas halladas <i>modi</i> valor real de las mitosis cerradas. Las mitosis cerradas son revisadas de nuevo en esta rutina. <i>posn</i> contiene la posicion de arch desde la cual comienzan los nucleos y las mitosis cerradas.	char 0 sin error 1 error	FILE *arch long posn int *dato int *modi	Usada en accion_mito y en only. Usa variables globales cr_cer, cr_ab y est.	C	MITSUBC3.C
recCromosomasCerrados	funcion que reconoce los cromosomas cerrados. <i>no</i> indica un contorno no existente (marco) <i>arch</i> apuntador a archivo temporal que sirve para almacenar los nucleos hallados en una pantalla. <i>dato</i> contiene el numero de mitosis cerradas halladas <i>error</i> > 0 si hubo error en la segmentacion = 0 no hubo error en la segmentacion	char 0 sin error 1 error	int *no FILE *arch int *dato char *error	Usada en accion_mito y en only. Usa variables globales cr_cer, cr_ab y est.	C	MITSUBC3.C

recNucleosEstimulados	<p>funcion que reconoce todos los nucleos estimulados y algunas mitosis cerradas.</p> <p><i>no</i> indica un contorno no existente (marco)</p> <p><i>arch</i> apuntador a archivo temporal que sirve para almacenar los nucleos hallados en una pantalla.</p> <p><i>dato</i> contiene el numero de nucleos hallados</p> <p><i>daton</i> contiene el numero de mitosis cerradas hallados</p> <p><i>posn</i> posicion del archivo <i>arch</i> desde donde se grabaron los actuales resultados de pantalla.</p>	<p>char</p> <p>0 sin error</p> <p>1 error</p>	<p>int no</p> <p>FILE *arch</p> <p>int *dato</p> <p>int *daton</p> <p>long *posn</p>	<p>Usada en accion_mito y en only.</p> <p>Usa variables globales <i>cr_cer</i>, <i>cr_ab</i> y <i>est</i>.</p>	C	MITSUBC3.C
retira	<p>retira todos los nodos de la lista lineal doblemente ligada usada para la impresion de un menu, en pantalla, de archivos disponibles.</p> <p><i>ap</i> apuntador movil sobre la lista</p> <p><i>ini</i> apuntador que marca el inicio de la lista</p> <p><i>fin</i> apuntador que marca el fin de la lista</p>	NADA	<p>node **ap,**ini,**fin</p>	<p>Debe ser usada para liberar memoria despues de usar la lista lineal doblemente ligada creada por inserta.</p>	C	MITSUBC5.C
rm	<p>Manda instruccion de medicion de limites al controlador MC2000</p>	<p>char</p> <p>0 sin error</p> <p>1 error</p>	VACIO	<p>Requiere del controlador MC2000 y del puerto serial COM1</p>	C	MC2000.C
segmenta	<p>segmenta objetos en la pantalla bajo un valor de gris umbral .</p>	<p>char</p> <p>0 segmento sin error</p> <p>1 error al realizar la segmentacion</p>	<p>int umbral</p>	<p>Usa MGRAPH, SUPERM2, SUPERM, LITPOS. Modifica SEUIL, SEUILS, YPAS, NPLAN, NCOU Puede ser usada siempre que se quiera segmentar.</p>	C	MITSUBC5.C
seg_new	<p>funcion que segmenta, inicializando siempre las tablas de BIOCOM para los contornos.</p>	<p>int</p> <p>regresa el valor al cual segmento</p>	<p>char elec,sharp</p> <p>char *error;</p> <p>int umbral0;</p>	<p>Modifica a DENS y las tablas de contornos.</p> <p>Depende de:</p> <p>INICON</p> <p>FINACQ</p> <p>MESSA</p> <p>segmenta</p> <p>find umbral</p>	C	MITSUBC5.C

selftest	Manda la Instruccion de auto prueba al controlador MC2000.	char 0 sin error 1 error	int *con int *axis con < 0 error del controlador axis < 0 error en la medicion de los ejes	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
setjoyvel	Ajusta la velocidad del joystick del controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
setlimit	inicia los limites de barrido	char 0 sin error 1 error	coord3 *ap	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
setpos000	Indica la posicion de origen del controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
setvel_fast	Pone en velocidad alta a los motores del controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
setvel_slow	Pone en velocidad baja a los motores del controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
sharp1	Primer metodo de afinamiento de la imagen	NADA	VACIO	Usa la funcion fg_sh1 de MATROX	C	MITSUBC5.C
sharp2	Segundo metodo de afinamiento de la imagen	NADA	VACIO	Usa la funcion fg_sh2 de MATROX	C	MITSUBC5.C
tran_palabra	transmite una cadena de caracteres por puerto serie	char 0 sin error 1 error	char *palabra	Requiere de dos.h y del puerto serial COM1	C	MC2000.C
unit	Pone las unidades de medida en mm en el controlador MC2000	char 0 sin error 1 error	VACIO	Requiere del controlador MC2000 y del puerto serial COM1	C	MC2000.C
valores_portaobj	funcion que sirve para imprimir valores de los nucleos estimulados e, las mitosis abiertas a y mitosis cerradas c en cada portaobjetos num.	NADA	int mm float e,a,c	Usada en _fortran mit fortran mitcmd	C	MITSUBC2.C

valores_toma	funcion que sirve para imprimir valores de los nucleos estimulados e, las mitosis abiertas a y mitosis cerradas c en cada toma.	NADA	float e,a,c	Usada en accion_mito	C	MITSUBC2.C
vec_asoc	actualiza el vector de asociados[]. control indica el control asoc indica el experimento asociado	NADA	int control,asoc	Modifica asociados[control] segun el valor de asoc	C	MITSUBC4.C
verify	funcion que comprueba la ultima ejecucion del indice mitotico. Interfaz con FORTRAN	NADA	VACIO	Usa y modifica el archivo NUCLEOS.REC por medio de pone, quita, changes y arch update	C	MITBASC6.C

Apéndice D

Diccionario de Variables y Constantes.

Estructuras Globales

Nombre	Componentes	Tipo
est	area,dgris,forma,cph	Estructura bounds
cr_ab	dgris,area,cph,cpv,ch,cv, porc,pp	Estructura bounds Estructura bounds
cr_cer	area,ngri,dgris,porc,cph, cpv,ch,cv,dif	Estructura bounds Estructura bounds
coord	x,y	Reales de 4 bytes
puntos	p00,p11	Estructura coord
coord3	x,y,z	Reales de 4 bytes
nodo_obj	tipo pixel marca	Entero de 2 bytes Estructura coord Caracter
dato1	x,y	Entero de 2 bytes
dato2	prom,dsv1	Reales de 4 bytes
dato3	area,forma,dgris,cph	Estructura dato2
dato_ab	area,dgris,cph,cpv,ch,cv, porc,pp	Estructura dato2 Estructura dato2
dato_cer	area,ngri,dgris,cph,cpv,ch, cv,porc,dif	Estructura dato2
node	liga_der,liga_izq posx,posy nombre[13]	Apunadores a Estructura node Caracter sin signo Cadena de caracteres sin signo
dato4	posx,posy nombre[13]	Caracter sin signo Cadena de caracteres sin signo
dato5	suma,sum_sqr	Real de 4 bytes
datos_prom	area,forma,dgris,cph int n	Estructura dato5 Entero de 2 bytes
datos_prom0;	area,ngri,dgris,cph,cpv ch,cv,porc,dif int n	Estructura dato5 Estructura dato5 Entero de 2 bytes
datos_prom1	area,dgris,cph,cpv,cli,cv porc,pp n	Estructura dato5 Estructura dato5 Entero de 2 bytes
acum	estim cromo_cer cromo_ab	Estructura datos_prom Estructura datos_prom0 Estructura datos_prom1
dato6	estim,cromo_ab cromo_cer	Real de 4 bytes
bounds	min,max	Real de 4 bytes
dato7	mar x,y ap,up	Caracter sin signo Real de 4 bytes Caracter
dato8	marca x,y	Entero de 2 bytes Real de 4 bytes
nod_pila	liga x0,y0,rx,ry n	Apuntador a estructura nod_pila Entero de 2 bytes Caracter sin signo

Variables Globales:

Variable	Tipo	Tamaño	Proposito	Valor Inicial
entrenamiento_activo	Cadena[13]	1 byte	Verifica si hay un entrenamiento activo	Segun lectura
offset_lam	Real	4 bytes	Contiene el	9881.26
off_z	Real	4 bytes	Contiene la posicion actual del controlador en z.	0.0
off_z.l	Real	4 bytes	Contiene la posicion actual del controlador en z.	0.0
maximo_y	Real	4 bytes	Guarda la posicion maxima alcanzable por las platinas en el eje y	Segun lectura
minimo_y	Real	4 bytes	Guarda la posicion minima alcanzable por las platinas en el eje y	Segun lectura
maximo_x	Real	4 bytes	Guarda el tamaño del barrido en X para todas las laminillas al tener un barrido predefinido	Segun lectura
desp_x	Real	4 bytes	Guarda el valor del desplazamiento en X necesario para cambiar de laminilla	113173.9024
porcent	Real	4 bytes	Guarda el valor del porcentaje minimo del indice mitotico	PORCENTAJE
pantalla	Real	4 bytes	Guarda el numero de pantallas que lee al calcular el indice mitotico	Se inicia al comenzar el calculo
porcy	Real	4 bytes	Guarda el porcentaje de muestreo de la laminilla en Y	
BAN1	Entero	2 bytes	Bandera que habilita el uso de archivos CMD y el movimiento de las platinas	0
BAN2	Entero	2 bytes	Bandera que habilita el uso de archivos CMD, calculo del indice mitotico y el movimiento de las platinas	0
AC	Entero	2 bytes	Bandera que habilita el uso de archivos CMD, calculo del indice mitotico y el movimiento de las platinas. Verifica si se llevo a cabo el	0
acon	Entero	2 bytes	Contiene el tipo de adquisicion	0
boton	Entero	2 bytes	Sirve para obtener el boton del raton	Cambia segun el boton del raton
num_portaobj	Entero	2 bytes	Contiene el numero de laminillas a revisar	1
lim_cnf_g	Entero	2 bytes	Limite de movimientos del eje Z al enfocar grueso	segun objetivo
lim_cnf_f	Entero	2 bytes	Limite de movimiento del eje Z al enfocar fino	segun objetivo
modo	Entero	2 bytes	Copia imagen de planos	2
mode	Entero	2 bytes	Copia imagen y grafico de planos	1
num_nucleos	Entero	4 bytes	Numero de cuerpos a contar al calcular el indice mitotico	2000

asociados	Cadena sin signo	8 bytes	Vector que indica que laminillas son experimentos asociados a que experimentos	{0,0,0,0,0,0}
controles	Cadena sin signo	8 bytes	Vector que indica que laminillas con controles	{69,69,69,69,69,69,69,69}
objetivo existe	Caracter sin signo	1 byte	Indica si esta definido algun objetivo	Segun lectura
objetivo definido	Caracter sin signo	1 byte	Indica si existe objetivo	Segun lectura
entrena existe	Caracter sin signo	1 byte	Indica si existe entrenamiento	Segun lectura
overlap existe	Caracter sin signo	1 byte	Indica si existe traslape de tomas	Segun lectura
xyzg	Estructura coord3		Contendra la posicion x,y,z	Segun uso
faxy	Estructura coord		Contiene el tamaño de la pantalla segun objetivo elegido	Segun lectura
veces[8]	Arreglo de estructura dato1		Contiene las veces que se barrera la pantalla en e y para cada laminilla	Segun calculo
estimul	Estructura dato3		Contiene los valores de los parametros para los nucleos estimulados	Segun lectura
romo_ab	Estructura dato_ab		Contiene los valores de los parametros para las mitosis abiertas	Segun lectura
romo_cer	Estructura dato_cer		Contiene los valores de los parametros para las mitosis cerradas	Segun lectura
overlap	Estructura dato8		Indica el valor del traslape en x e y para cada toma y el objetivo definido	{-1,0,0,0,0}
region[8]	Arreglo de estructura puntos		Contiene las coordenadas sup. derecha e inf. izquierda del rectangulo de barrido para cada laminilla	Segun lectura
mv	Entero corto	2 bytes	Variable a usar con MESSA	1
mc	Entero corto	2 bytes	Variable a usar con MESSA	1
nummens	Entero corto	2 bytes	Contendra el numero de mensaje a desplegar	Segun uso
plano2	Entero corto	2 bytes	Contiene el valor para identificar al plano numero dos de la MATROX	PLANO2
consel	Entero corto	2 bytes	Contendra el contorno elegido	Segun uso

Variables Locales:

Variable	Tipo	Tamaño	Proposito	Archivo
reg	Union REGS definida en bios.h		Manejo de interrupciones del BIOS desde C	MC2000.C
status	Entero sin signo	2 bytes	Recibe el estatus de las funciones del bios.h	MC2000.C
*resul	Apuntador a estructura nod_redo		Apuntador movil de la lista de resultados de cada laminilla	MITBASC6.C,MITSUBC8.C
*aresul	Apuntador a estructura nod_redo		Apuntador final de la lista de resultados de cada laminilla	MITBASC6.C,MITSUBC8.C
*iresul	Apuntador a estructura nod_redo		Apuntador inicial de la lista de resultados de cada laminilla	MITBASC6.C,MITSUBC8.C
*ins	Apuntador a estructura nod_red0		Apuntador movil de la lista de inserciones al archivo NUCLEOS.REC	MITBASC6.C,MITSUBC8.C
*ains	Apuntador a estructura nod_red0		Apuntador final de la lista de inserciones al archivo NUCLEOS.REC	MITBASC6.C,MITSUBC8.C
*iins	Apuntador a estructura nod_red0		Apuntador inicial de la lista de inserciones al archivo NUCLEOS.REC	MITBASC6.C,MITSUBC8.C
*del	Apuntador a estructura nod_pil		Apuntador movil de la lista de bajas al archivo NUCLEOS.REC	MITBASC6.C,MITSUBC8.C
*adel	Apuntador a estructura nod_pil		Apuntador final de la lista de bajas al archivo NUCLEOS.REC	MITBASC6.C,MITSUBC8.C
*idel	Apuntador a estructura nod_pil		Apuntador final de la lista de bajas al archivo NUCLEOS.REC	MITBASC6.C,MITSUBC8.C

*archivo	Apuntador a estructura de archivo definida en stdio.h		Manejo de Archivo de control NUCLEOS.REC	MITBASC6.C,MITSUBC8.C
dato6 pant	Estructura dato6		Manejo de valores de Nucleos, Mitosis cerradas y abiertas en cada pantalla	MITBASC6.C,MITSUBC8.C
actual	Entero	2 bytes	Indica el portaobjetos que se revisa	MITBASC6.C,MITSUBC8.C

Constantes

Nombre	Tipo
AREA MAX	Real
ERR INF 0.98	Real
ERR SUP	Real
INC C	Real
MAX DESP POBJ X	Real
PI	Real
PORCENTAJE	Real
REDUC	Real
INC G	Entero
INC P	Entero
INICIO	Entero
LIM CR	Entero
LINEAS	Entero
MAXCONTROLES	Entero
ESQY	Entero
ESQX	Entero
DISTAN 7	Entero
COM2 1	Entero
DESP 5	Entero
DESPX 20	Entero
DESPY 20	Entero
DIST PIX 100	Entero
ABIERTOS 4	Entero
OPENED 1	Entero
CERRADOS 3	Entero
GRANDES 1	Entero
PEQUENOS 2	Entero
NULO 0	Entero
NEUTRO 5	Entero
APROX POL 1	Entero
MAX GRIS	Entero
MED	Entero
MEDIA	Entero
MOV	Entero
MOV G	Entero
PAG	Entero
PLANO	Entero
PLANO2	Entero
PREC FLOAT 8	Entero
RADIO NUG 4	Entero
RADIO NUC 2	Entero
RADIO AB 1	Entero
RADIO CER 1	Entero
SIMB1	Entero
SIMB2	Entero
SIMB6	Entero
SIMB4	Entero
SIMB3	Entero

SIMB5	Entero
TAB	Entero
VECES	Entero
XMIN SCREEN	Entero
YVENI SCREEN	Entero
YVEN SCR	Entero
ARCHIVO DESP	Cadena
ARCHIVO OBJ	Cadena
ARCHIVO PAR	Cadena
ARCHIVO OVER	Cadena
NAME	Cadena
NAME0	Cadena
NAME1	Cadena

Constantes Locales

Nombre	Tipo	Archivo
SERIAL2	Hexadecimal	MC2000.C
SERIAL1	Hexadecimal	MC2000.C
INT14	Hexadecimal	MC2000.C
BAUD 4800	Hexadecimal	MC2000.C
BAUD 1200	Hexadecimal	MC2000.C
TAM PAL	Hexadecimal	MC2000.C
NO PAR	Hexadecimal	MC2000.C
BIT STP	Hexadecimal	MC2000.C
DIST	Entero	MITONLYC.C
DIST1	Entero	MITONLYC.C
DEMORA	Real	MITSUBC3
RAD X	Entero	VERIFY.H,MITSUBC8.C
RAD Y	Entero	VERIFY.H,MITSUBC8.C
VIDEO	Hexadecimal	MITSUBC1.C
LX	Entero	MITSUBC1.C
MAX IT	Entero	MITSUBC1.C

Apéndice E

Glosario

PDI Procesamiento Digital de Imágenes.
Pixeles Pixels abreviación de "Picture Elements" en ingles. Los bits en la posición (x,y) del buffer de pantalla que representan la información de una imagen.
LUT Look Up Table en ingles. Tablas de asignación.
MC2000 Sinónimo de MultiControl 2000. Controlador de las platinas y perilla de enfoque para el microscopio.
Threshold Umbral.
Thresholding Método para hallar un umbral para segmentar.

Apéndice F

Interfaz C-FORTRAN

INTERFAZ FORTRAN-C

C

Para hacer una llamada de una rutina en FORTRAN desde C se declara lo siguiente:

```
extern tipo_de_retorno_fortran Nombre(Lista_argumentos);
```

Cuando se quiere crear una rutina en C que pueda ser llamada desde FORTRAN se define en el archivo en lenguaje C como sigue:

```
tipo_de_retorno_fortran Nombre(Lista_argumentos)
```

FORTRAN

Para hacer una llamada de una rutina en C desde FORTRAN se declara lo siguiente:

```
INTERFACE TO { tipo_de_retorno FUNCTION | SUBROUTINE } [C, ALIAS:'nombre_funcion_C']  
parámetros
```

```
END
```

donde **_fortran**, **extern**, **FUNCTION**, **C**, **INTERFACE TO**, **SUBROUTINE**, **END** y **ALIAS** son palabras reservadas en Microsoft C y FORTRAN respectivamente.