



COORDINACIÓN DE SERVICIOS  
DOCUMENTALES - BIBLIOTECA

UNIVERSIDAD AUTÓNOMA METROPOLITANA IZTAPALAPA  
DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

---

*DETERMINACIÓN DE LA DENSIDAD  
ANULAR DE CULTIVOS CON CRECIMIENTO  
HOMOGÉNEO POR TÉCNICAS DE  
PROCESAMIENTO DIGITAL DE IMÁGENES*

227465

TESIS PARA OBTENER EL GRADO DE  
MAESTRÍA EN INGENIERÍA BIOMÉDICA

**MERCEDES JATZIRI GAITÁN GONZÁLEZ**

**ASESORA: M. EN I. SONIA CHARLESTON VILLALOBOS**

México, D. F., enero 1993

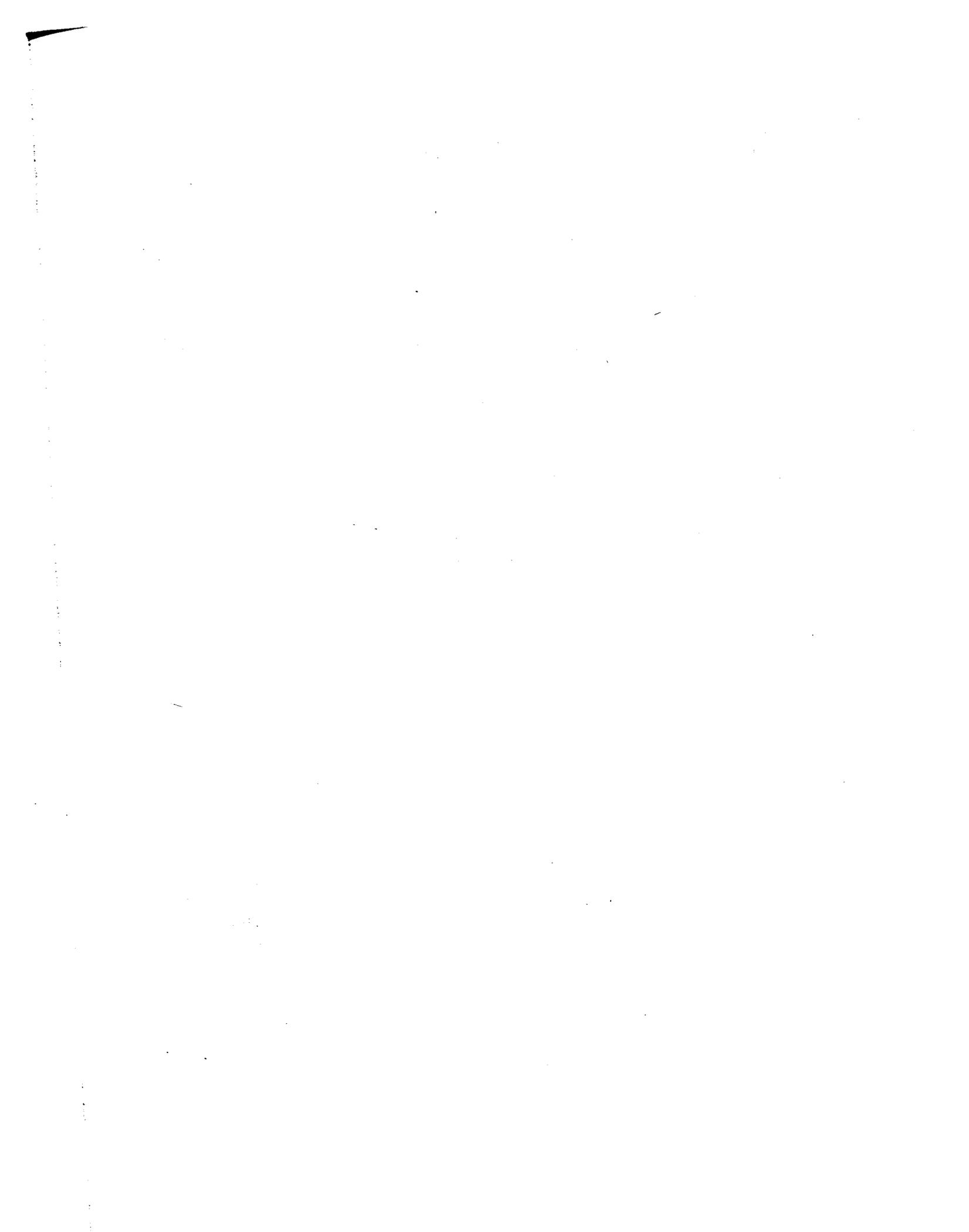
17/06/02

Se agradece al Consejo Nacional de Ciencia y Tecnología el apoyo brindado para la realización de estos estudios de posgrado. Asimismo, se agradece al Area de Microbiología del Depto. de Biotecnología, UAM-Iztapalapa, por su ayuda y disponibilidad en el cultivo de los microorganismos y la adquisición de las imágenes.

## TABLA DE CONTENIDO

<b>INTRODUCCION.</b>	1
<b>ANTECEDENTES.</b>	4
<b>PLANTEAMIENTO DEL PROBLEMA.</b>	8
Bases biológicas.	9
Justificación.	10
<b>HIPOTESIS Y OBJETIVOS.</b>	12
<b>METODOS.</b>	13
Criterios de diseño	13
Descripción del sistema desarrollado	14
1. Biblioteca básica de funciones.	15
a) Unidad General	17
b) Unidades de manejo del sistema	18
c) Unidades de procesamiento de la imagen	22
2. Determinación y análisis de la densidad anular de cultivos	31
a) Localización del centro del cultivo	32
b) Determinación del radio del cultivo	37
c) Cálculo de la densidad anular	37
d) Unidades de procesamiento de la densidad anular	39
3. Programa de usuario	40
4. Estandarización del proceso de adquisición	42
Procedimientos de prueba del sistema.	44
<b>RESULTADOS.</b>	48
Biblioteca de funciones y unidades de análisis	48
Calibración	48
Densidad anular del cultivo	48

<b>ANALISIS Y DISCUSION DE RESULTADOS.</b>	54
Biblioteca de funciones.	54
Estandarización de la adquisición.	55
Densidad anular del cultivo.	57
Ejemplo de aplicación de la densidad anular.	67
<b>CONCLUSIONES Y PERSPECTIVAS.</b>	70
<b>ANEXO 1.</b> Código de los archivos de definición de los procedimientos incluidos en la biblioteca general de funciones.	72
<b>ANEXO 2.</b> Programación desarrollada para la obtención y el análisis de la densidad anular de cultivos con crecimiento homogéneo.	86
<b>BIBLIOGRAFIA.</b>	106



# INTRODUCCION

Las imágenes han sido ampliamente utilizadas como fuente de información en muchas disciplinas; inicialmente sólo en forma cualitativa y posteriormente como parte de un proceso cuantitativo de extracción y análisis de información. En la metalurgia (33), la biología (27,30), la biomecánica (18,28), la medicina (29), la fisiología (44), la antropología física (50) y otras disciplinas, las mediciones en fotografías obtenidas por diversas técnicas, o las mediciones a través de microscopios calibrando con rejilla, han sido consideradas como una herramienta útil tanto en el campo de la investigación como para aplicaciones diversas.

Tradicionalmente, las mediciones y el análisis se han realizado sobre imágenes impresas (fotografías, placas radiográficas, proyecciones, etc.) o directamente sobre el objeto a medir; muchas veces con la ayuda de microscopios u otros instrumentos que permiten la visualización del objeto o fenómeno a analizar.

El desarrollo de microscopios con mayores capacidades y mejores y más variadas características ha permitido una evolución importante en el análisis y cuantificación microscópico de materiales y organismos (24,46). Por otro lado, las técnicas de tinción, el uso de filtros ópticos, medios de contraste, retículas calibradas, etc. han facilitado este tipo de análisis, sin llegar a un punto de simplificación suficiente para dejar al proceso exento del manejo manual y de la dependencia de la experiencia de los observadores.

Por otro lado, se han generado numerosos instrumentos cuya salida es una imagen y que permiten visualizar diversas características de una escena o fenómeno. Entre estos instrumentos están los sistemas de rayos X, las gamacámaras, los equipos de ultrasonido, y los sensores ópticos de gran capacidad y con respuesta a diferentes espectros.

Así como el desarrollo de instrumentos de observación ha impulsado la utilización de las imágenes en diversas ciencias, el diseño de sistemas de adquisición de imágenes y de computadoras cada vez más rápidas, de mayor capacidad de procesamiento y de menor costo ha facilitado que más técnicas de procesamiento de imágenes sean desarrolladas y utilizadas en un gran número de aplicaciones y laboratorios.

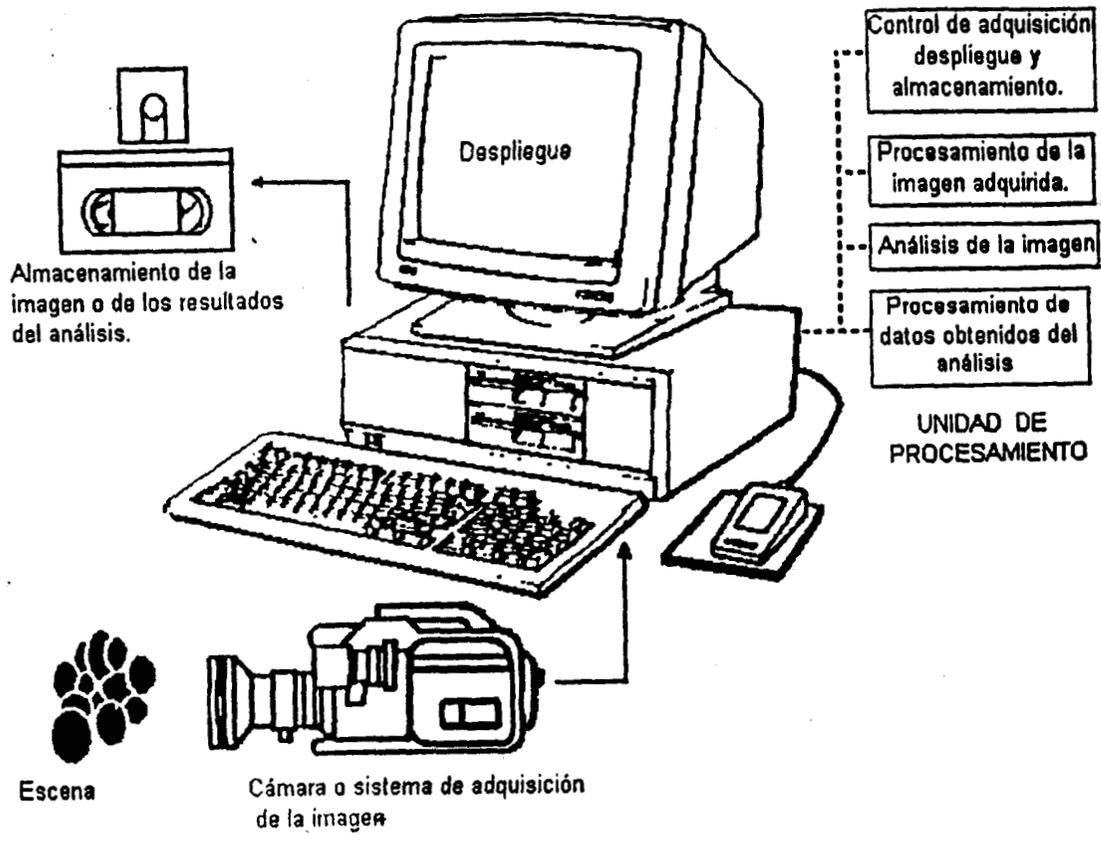
Con toda esta tecnología empleada en el manejo y procesamiento de imágenes, el uso de la fotometría se ha extendido y se le ha reconocido como una importante ayuda con muchas más aplicaciones.

En general, la digitalización de señales permite su tratamiento matemático en forma computarizada para extraer, cuantificar y clasificar la información de interés. Una imagen es considerada como una señal bidimensional que refleja la distribución espacial de la respuesta espectral (variación de la energía radiante en función de su longitud de onda) de los elementos que componen la escena (26). Mediante el procesamiento de tal señal es posible mejorar sus características de apariencia, realizar transformaciones y realces que permitan resaltar sus características importantes y extraer la información relevante. Por otro lado, muchos de estos procedimientos pueden realizarse automáticamente o tan solo con la supervisión de un operador.

Por estas razones, la utilización de técnicas de procesamiento digital de imágenes ha hecho posible la creación de nuevas metodologías de análisis y la simplificación para el usuario de las ya existentes. Lo anterior ha permitido una cuantificación precisa de aquello que anteriormente era cualitativo o bien que su medición era producto de largas horas de observación, con una gran dependencia de la experiencia y disposición de la persona que realizara el estudio.

El análisis de una imagen consiste en su descripción en términos de las propiedades de uno o varios elementos de la escena, así como de las relaciones entre éstos. El resultado final del análisis de imágenes puede ser otra imagen, una lista de características morfológicas, un campo vectorial que represente posiciones o velocidades de diversos elementos, un mapa que represente las regiones de la imagen o un arreglo que describa ciertos parámetros extraídos de dicha imagen. En la figura 1 se muestra el sistema completo requerido para el análisis de imágenes.

Desafortunadamente, el proceso de adquisición de una señal implica una pérdida de información y una degradación de la misma (26), originadas por la digitalización de las tonalidades de la imagen, las limitaciones físicas de los sistemas de adquisición, deficiencias de los sensores y problemas geométricos en la adquisición, por citar algunas causas. Para procesos específicos de obtención de información a partir de imágenes, esta pérdida o



**FIGURA 1.** Sistema general para el análisis de imágenes por técnicas de procesamiento digital.

degradación puede no ser importante. Por otro lado, existen técnicas que facilitan la extracción de la información requerida (15,26). Dichas técnicas se pueden clasificar como de realce, de transformación y de restauración y pueden realizarse ya sea en el dominio espacial o en el de la frecuencia.

## **ANTECEDENTES**

El procesamiento digital de imágenes ha sido empleado ampliamente en las ciencias biológicas. Específicamente en la medicina, la utilización de imágenes adquiridas de diversas maneras ha aumentado el número de técnicas no invasivas para el estudio de muchos padecimientos que requieren de la medición de las características geométricas o de absorción de alguna parte del cuerpo o de cierto órgano (1). Así mismo, ha conducido al desarrollo de procesos de cuantificación de fenómenos y características de los órganos (39,48) y ha permitido desarrollar nuevas metodologías clínicas y mejorar las ya existentes (47,49).

Otras aplicaciones biomédicas en las que el procesamiento digital de imágenes ha sido utilizado son en biomecánica para el análisis del movimiento humano (32); en histología, para el análisis de cortes (3); en microbiología, en pruebas de susceptibilidad bioquímica y antimicrobiana (11); en inmunología, para el análisis morfológico de micrografías de preparaciones con oro (5); en hidrobiología, para la determinación del volumen (13) o la biomasa de bioplancton (4) y la clasificación de muestras de zooplancton (22); en biología celular, para el análisis y la cuantificación de células (10) y la comparación morfológica de sus núcleos (21); en genética, para la clasificación semiautomática de cromosomas (14) y su estudio de transferencia de material genético (17); en bacteriología, para realizar la cuenta de bacterias (37), etc.

En particular, en el campo de la biotecnología el uso de las técnicas de procesamiento digital de imágenes ha mostrado un incremento considerable en los últimos años.

La biotecnología se puede definir como la aplicación intensiva y óptima de seres, sistemas o procesos vivos y sus derivados en la industria de servicios o de manufactura.

Entre los problemas que conciernen a la biotecnología se encuentran la obtención, la caracterización y el mejoramiento de cultivos de diversos organismos utilizados en la producción de sustancias para la industria alimenticia, farmacéutica, textil, etc.; la optimización industrial de la utilización de dichos organismos; la cuantificación de procesos microbianos y la obtención de modelos de crecimiento y movilidad.

Si bien existen numerosas aplicaciones, el empleo de técnicas de procesamiento digital de imágenes en biotecnología ha sido enfocado principalmente a la enumeración de células o partículas (10,37,45) y a la caracterización morfológica de micelios, mediante el uso de imágenes adquiridas a través de un microscopio ( 2,31,35,40,42).

**Aplicaciones para el conteo de células.** Pettipher y Rodrigues (37) describieron una técnica de control de calidad semiautomática para contar bacterias y células somáticas en la leche, utilizando un microscopio de epifluorescencia. Costello y Monk (10) utilizaron un método similar para contar rápidamente células de *Sccharomyces cervisae* encontrando que si bien el resultado dependía de la intensidad de iluminación, el contraste y el nivel de gris de segmentación, todos estos parámetros podían ser controlados, pero con la ventaja de que la medición no dependía de la concentración de etanol ni de azúcar, sustancias cuyas concentraciones varían significativamente durante la fermentación y que afectan la medición realizada por otros métodos; Sjollema y Cols. (45) realizaron un sistema para la enumeración en tiempo real de microorganismos adheridos en una placa colocada en forma paralela al flujo de aire.

**Aplicaciones para la caracterización morfológica de micelios.** La caracterización morfológica de microorganismos filamentosos es importante para el diseño y operación de las fermentaciones de dichos organismos. El enredo de las hifas puede originar suspensiones con características que causan problemas en la transferencia de masa y calor.

Antes de contar con técnicas de procesamiento de imágenes, sólo se realizaba un caracterización cualitativa o mediciones de longitud y diámetro por observación directa a través del microscopio (31). Los parámetros de caracterización seleccionados fueron las longitudes de la hifa principal, total de las hifas de una partícula, media de las ramas y de los segmentos, número de ramificaciones y el diámetro de las hifas. La evolución en la utilización del procesamiento digital de imágenes en esta aplicación ha ido desde la medición computarizada a través de la proyección de una fotografía, usando una tabla digitalizadora (31), hasta la semiautomatización del proceso mediante la adquisición de la imagen microscópica, su procesamiento (filtrado, segmentación, esqueletización) y la medición

automática de los parámetros ya mencionados, todo supervisado y con la posibilidad de edición por parte de un operador (35).

Packer y Thomas (35) realizaron mediciones morfológicas microscópicas en cultivos de *Aspegillus niger* y *Penicillum* en la forma manual y la semiautomatizada, comparando tiempos y características de precisión. Encontraron que al utilizar el procesamiento digital de imágenes se lograban mejores resultados en periodos de tiempo más cortos y con la posibilidad de determinar otros parámetros. Aparentemente, la programación realizada por estos investigadores es específica para el problema planteado, si bien utiliza algoritmos generales del procesamiento de imágenes.

A los parámetros para la caracterización morfológica de los microorganismos, Yang y Cols. (54) agregaron el ángulo de crecimiento de la punta o ángulo de ramificación con el fin de caracterizar la dirección del crecimiento de una hifa o rama en un plano. Mediante sus resultados plantearon un modelo del crecimiento direccional de las ramas, mismo que complemento al modelo general planteado por los mismos autores (53).

Dado que muchos organismos filamentosos tienden a formar agregados en forma de perdigones o mezclas de micelios dispersos y agregados, con el fin de encontrar la morfología para lograr un producto óptimo, Reich y Cols. (41) utilizaron el procesamiento de imágenes para realizar la caracterización de la morfología de dichos agregados, mediante la medición de la frecuencia de distribución de los agregados o de los micelios filamentosos, el área media de los agregados o de los micelios filamentosos, el porcentaje de área ocupado por los agregados y un índice de forma de dichos agregados. Este último parámetro lo determinaron como la suma de la diferencia de las áreas de cuatro imágenes resultantes de aperturas consecutivas con elementos estructurales (cuadrados) de tamaño creciente entre el número de aperturas. Además de las operaciones de apertura, las herramientas de procesamiento de imágenes empleadas fueron la segmentación, otros filtros morfológicos, discriminación de agregados y micelios por nivel de gris y determinación de contornos.

Packer y Cols. (34) utilizaron programación para la caracterización morfológica de microorganismos filamentosos (35), agregándole otras rutinas de procesamiento (segmentaciones, cierres, dilatación, sustracción de imágenes, medición de diámetros) para realizar la diferenciación de regiones degeneradas y sanas de la hifa y estimar su volumen y biomasa.

**Otras aplicaciones.** Camp y Sublette (8) utilizaron un sistema de visión computarizado para detectar las condiciones de estado de un bioreactor, basandose en el cambio de color que ocurre debido a la formación de un precipitado coloidal de sulfuro.

Mediante el procesamiento digital de imágenes se han realizado diversos estudios de estimación de volumen y biomasa, principalmente de bioplacton (4,13).

Para la caracterización de poblaciones de levaduras, Huls y Cols. (20) propusieron un sistema que combinara al contador electrónico de partículas, la citometría de flujo y el análisis bidimensional de imágenes. Los dos primeros procedimientos permiten asegurar el estado estable del crecimiento y el análisis digital de la imagen, proporciona información adicional como el patrón de aumento de volumen de células hijas y padres, el coeficiente de variación del tamaño de las células en la división y otras. Estos autores plantearon un procedimiento diferente para determinar el volumen de la célula a partir de la rotación del plano conocido, ya que anteriormente el volumen se calculaba mediante el eje mayor y menor del organismo, considerandolo un elipsoide.

Entre las aplicaciones en biotecnología de la densidad óptica, representada en la imagen digitalizada como el nivel de gris, se encuentran la detección automática de transferencia de material genético (17), la diferenciación de material aglutinado y micelios filamentosos en observaciones microscópicas de bacterias y hongos (41), el reconocimiento de las zonas degeneradas de las hifas (34) y el análisis de la movilidad aleatoria de bacterias (12).

La mayoría de las aplicaciones en biotecnología del procesamiento digital de imágenes han sido encaminadas a la caracterización microscópica de organismos; en muy pocas aplicaciones se ha realizado un análisis de imágenes macroscópicas de los cultivos. Sin embargo, para realizar la caracterización completa de un microorganismo ésta no puede realizarse sólo en forma microscópica, ya que las características que pueden observarse a esa escala difieren de las observables a nivel macroscópico. Ambas observaciones pueden considerarse complementarias para el análisis completo de las características morfológicas y de crecimiento de un microorganismo bajo determinadas condiciones de estudio.

## PLANTEAMIENTO DEL PROBLEMA

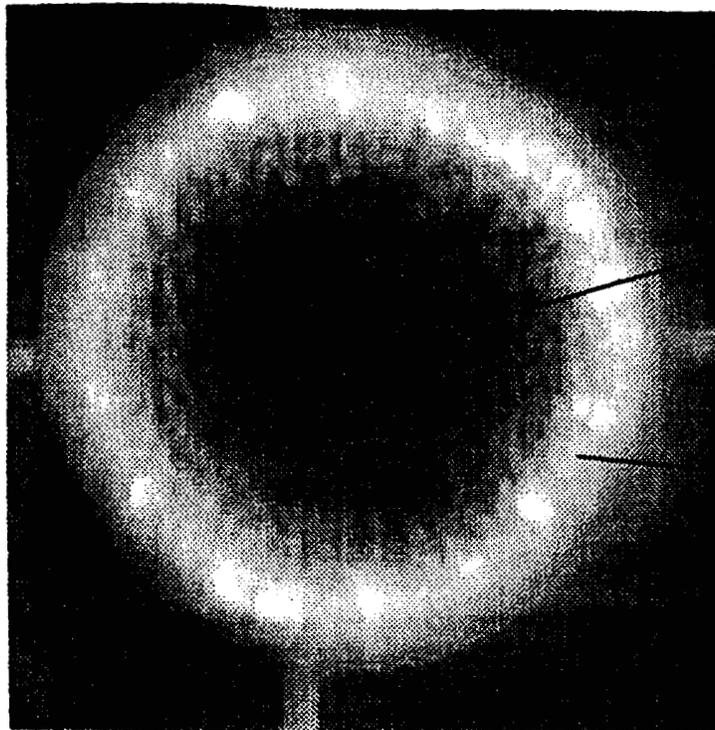
En el Area de Microbiología del Departamento de Biotecnología, de la Universidad Autónoma Metropolitana-Iztapalapa, actualmente se están realizando investigaciones encaminadas a la caracterización de diferentes mutaciones de microorganismos cultivados en diversos medios. Estas investigaciones están encaminadas a la diferenciación y clasificación de cepas de diferentes mutaciones de un mismo microorganismo, para localizar aquellas con las características necesarias para una aplicación específica. Particularmente, se han venido realizando investigaciones sobre un hongo, el *Aspergillus niger*.

Entre los microorganismos más estudiados en biotecnología, por su amplia utilización en la industria alimenticia, textil y en la producción de farmacéuticos, se encuentran diversas clases de *Penicilium*, de *Aspergillus* y los *Streptomitos*. Estos tres tipos de microorganismos crecen formando micelios y han sido estudiados morfológicamente con las técnicas de procesamiento de imágenes ya descritas (2,31,35,40,42).

En particular, el *Aspergillus niger* es un hongo que generalmente crece en forma homogénea y es ampliamente utilizado en la industria alimenticia para la producción de ácido cítrico (43), en la textil y otras por su gran utilización en la producción de pectinasas (25) y en la farmacéutica para la producción de productos intermedios en la fabricación de ciertos medicamentos (19,23). También ha sido utilizado con fines ecológicos en la bioconversión de residuos de destilerías (16).

Los diferentes mutantes de este hongo presentan características diferentes que los hacen utilizables para una u otra aplicación. El estudio de dichas características y de su capacidad de desarrollo y crecimiento es indispensable para lograr una óptima producción de cada una de las aplicaciones específicas.

Para la obtención de características morfológicas y de desarrollo de estos microorganismos, y de otros que presenten crecimiento homogéneo, se planteó como una alternativa la obtención del perfil radial de densidad óptica del cultivo, mejor llamado densidad anular, a partir de imágenes macroscópicas de los cultivos inoculados de manera puntual.



Zona añeja y  
zona fructificante

Zona productiva  
y  
zona de extensión

**FIGURA 2.** Distribución de las zonas añeja, fructificante, productiva y de extensión en un cultivo de *Aspergillus niger* de una semana de edad.

### **Bases biológicas.**

En una inoculación puntual, los microorganismos de crecimiento homogéneo presentan una distribución circular alrededor del punto de inoculación. La colonia así formada tiene diferentes zonas con funciones y características específicas: la zona añeja, la fructificante, la productiva y la de extensión, distribuidas en ese orden del centro hacia afuera (51,52). Macroscópicamente, las zonas añeja y fructificante pueden diferenciarse de la productiva y la de extensión porque en las dos primeras existe cierto grado de esporulación que provoca un color negro y que se diferencia del color claro de las otras dos. Lo anterior se muestra en la figura 2 para un cultivo de *Aspergillus niger* de una semana de edad. Las zonas dependen

directamente del desarrollo del microorganismo y su distribución varía entre microorganismos diferentes (51) y aún dentro de una misma especie, entre mutaciones diferentes.

Debido a que se están considerando microorganismos con crecimiento homogéneo, las diferentes tonalidades que presente el cultivo y que correspondan a los diversos grados de desarrollo estarán distribuidas en forma de patrones anulares alrededor del centro. Los organismos que se desarrollan de esta manera, presentan por tanto simetría radial de su densidad óptica.

### **Justificación.**

Dado que tanto las características morfológicas de los microorganismos o de sus cultivos, como sus cambios temporales brindan información necesaria para su análisis y clasificación, el uso del procesamiento de imágenes para medir dichas características resulta una parte importante de un método que permita evaluar y caracterizar tanto a los organismos, como a sus procesos de crecimiento, con el fin de optimizar su selección. Este análisis se debe realizar tanto macroscópica como microscópicamente, ya que las características a medir no siempre son observables a cualquier escala. Por otro lado, es deseable realizar mediciones del mismo cultivo en instantes diferentes de tiempo para tratar de describir su proceso de crecimiento.

La tendencia hacia la automatización así como los requerimientos necesarios para cubrir las características ya mencionadas, hacen necesaria la utilización de métodos de análisis y cuantificación rápidos, sin descuidar la precisión, exactitud y repetibilidad deseables en cualquier medición.

El procesamiento de imágenes permite cubrir estos requerimientos, y si bien existen sistemas comerciales para realizar diversos procedimientos del procesamiento de imágenes, pocos son los que permiten la utilización de técnicas de realce y de tratamiento y manejo de la información para una aplicación específica, de manera que resulten en procedimientos que sean transparentes a un operador. Estos sistemas comerciales demandan del entendimiento por parte del usuario, de los procesos realizados sobre la

imagen, lo que provoca que su desempeño sea pobre o complicado para la solución de problemas específicos, que generalmente se deben realizar en forma manual en este tipo de sistemas.

Algunos sistemas comerciales permiten la utilización de macroinstrucciones que el usuario debe llamar en un programa para generar una aplicación específica (20). Una vez comprendido el problema y las herramientas de procesamiento que se requieren, estos sistemas facilitan la utilización de estas instrucciones para generar el programa final de la aplicación planteada. Es posible que en estos sistemas se requiera de programación adicional para cierto tipo de aplicaciones.

Contando con una biblioteca básica de funciones, la programación de aplicaciones específicas es más sencilla y permite adaptarse a las necesidades de los usuarios, manteniendo, para su comodidad, cierta parte del procesamiento transparente a ellos.

Dentro de los sistemas comerciales conocidos, ninguno permite la determinación de la densidad anular de cultivos con crecimiento homogéneo de manera sencilla para el usuario. Por tanto, el desarrollo de un sistema para determinar dicha densidad, surge como una necesidad específica planteada por el Departamento de Biotecnología y con posibles aplicaciones en numerosas investigaciones.

Por otro lado, el diseño del procedimiento particular puede partir de la generación de una biblioteca básica de funciones para el procesamiento digital de imágenes, a fin de crear una plataforma para el desarrollo de aplicaciones futuras con la posibilidad de interacción entre las aplicaciones desarrolladas en base a la misma biblioteca.

Es importante resaltar la necesidad de un grupo interdisciplinario para el desarrollo de programas de procesamiento de imágenes con aplicaciones específicas, para lograr la utilización óptima de las herramientas físicas y de procesamiento con que se cuenten.

## **HIPOTESIS**

En base a que:

- a). los cultivos de ciertos hongos y bacterias tienen un crecimiento homogéneo a partir de su punto de inoculación,
  - b). su desarrollo lo realizan en forma radial, y
  - c). los cultivos presentan densidades ópticas diferentes a nivel macroscópico para los diversos grados de desarrollo,
- es posible realizar un análisis semiautomático de las imágenes de este tipo de cultivos, que permita obtener la densidad anular del cultivo para caracterizarlo.

## **OBJETIVO GENERAL**

Desarrollar un sistema de procesamiento digital de imágenes que permita el análisis morfométrico macroscópico de cultivos microbianos en forma de colonias a partir de la determinación de su densidad anular.

## **OBJETIVOS ESPECIFICOS**

1. Generar una biblioteca de funciones para el procesamiento digital de imágenes, que incluya los principales métodos de realce de imágenes, de tal modo que permita su aplicación en diversos campos y facilite la generación de programas de procesamiento y análisis de imágenes para aplicaciones específicas.
2. Diseñar, desarrollar y programar los algoritmos necesarios para determinar el perfil radial o densidad anular de imágenes con simetría radial, tales como colonias de microorganismos con crecimiento homogéneo.

## METODOS

Para la exposición del sistema desarrollado, se especificarán primeramente los criterios de diseño empleados en su generación. Posteriormente se describirá cada parte del sistema, incluyendo la forma en que se realizó la normalización de la adquisición de las imágenes. Finalmente, se presentarán las pruebas a las que se sometió el sistema.

### **Criterios de diseño.**

Como las imágenes utilizadas en cada aplicación presentan características específicas, en el presente trabajo se planteó que el procesamiento de imágenes para resolver problemas de una rama del conocimiento específica debe abarcar:

1. Técnicas comunes de procesamiento de imágenes que permitan el realce general de la imagen y su mejoramiento global.
2. Técnicas de realce de la información específica relacionada con los problemas que presentan las imágenes de esta disciplina, y
3. Algoritmos específicos de tratamiento y manejo de información para los problemas particulares planteados.

En general, las aplicaciones reportadas en la literatura requieren de programación específica. Algunas están basadas en sistemas de procesamiento de imágenes comerciales, realizando consecutivamente instrucciones proporcionadas por el sistema. La mayoría son procedimientos semiautomáticos o supervisados por operador.

Las ventajas de los sistemas semiautomáticos sobre los manuales son la disminución de la dependencia del operador, la disminución del tiempo total de medición o análisis y la alta capacidad de procesamiento de información resultante. Algunos de los problemas que se pueden presentar tanto en este tipo de sistemas como en los totalmente automáticos son una baja capacidad de discriminación de niveles de gris y posibles dificultades en el análisis o detección por complejidad de la imagen (6). Estas dificultades son menos problemáticas en el caso de los sistemas semiautomáticos ya que existe la supervisión por parte del operador.

Dentro de esta filosofía, se desarrolló una biblioteca básica para el procesamiento de imágenes que incluye funciones generales para el manejo de la imagen, para el manejo del sistema de despliegue y procedimientos de realce de la imagen.

La biblioteca básica de funciones para el desarrollo de programación en el procesamiento digital de imágenes, debe incluir los algoritmos de uso general y la definición de las variables y estructuras de datos necesarias, de modo que sea posible utilizar estas unidades para el desarrollo de programación que permita resolver problemas específicos en forma más sencilla y rápida, incluyendo la ventaja de permitir en el momento que sea necesario, la interacción entre los diferentes programas desarrollados.

El resto de la programación se desarrolló bajo esta base y con la utilización de funciones de la misma. Para el programa de usuario, las características planteadas fueron su fácil operación y para ciertas rutinas, la posibilidad de supervisión.

El sistema fue programado en lenguaje C por la facilidad de interacción con la computadora. Está diseñado para trabajar en una microcomputadora con monitor VGA de color. La intención al utilizar esta plataforma fue permitir su operación en el sistema más básico posible, para facilitar su utilización en diversos laboratorios.

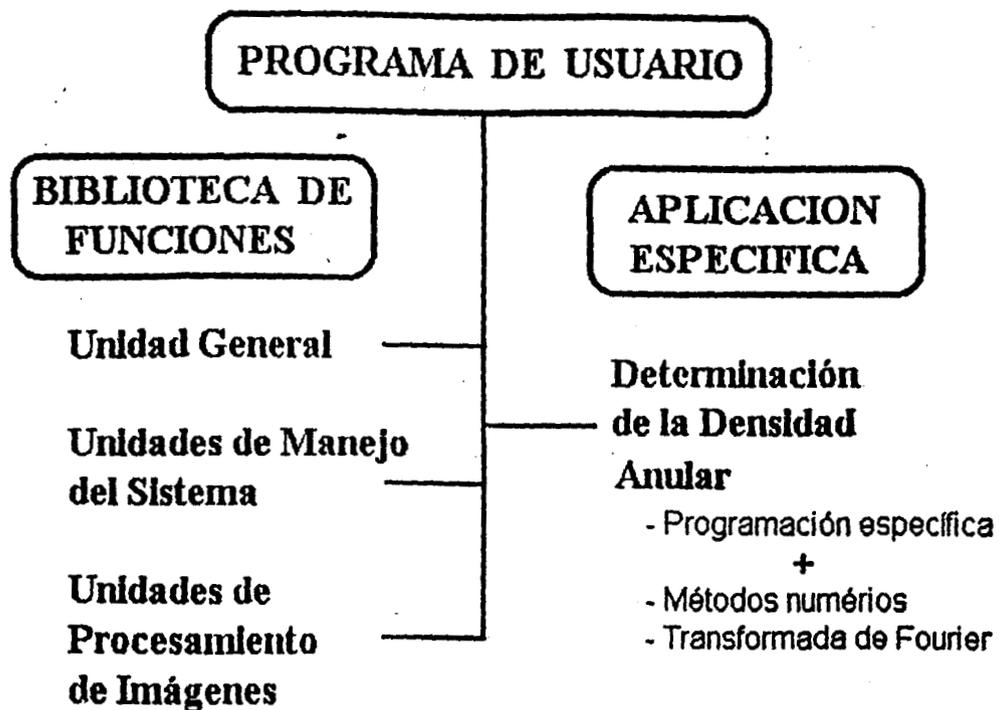
Se consideró que el rango dinámico de las imágenes sería de 256 niveles de gris, por ser el rango más utilizado en los sistemas de adquisición de imágenes disponibles.

### **Descripción del sistema desarrollado.**

El sistema desarrollado está formado por tres partes:

1. Una biblioteca básica de funciones.
2. Procedimientos para la determinación y análisis de la densidad anular de cultivos con crecimiento homogéneo.
3. Programa de usuario.

En la figura 3 se presenta un diagrama general del sistema desarrollado, mostrando la interrelación entre las partes del mismo.



**FIGURA 3.** Diagrama general del sistema desarrollado.

Para la descripción del sistema, se expondrán inicialmente las características, funciones y estructuras de datos incluidas en la biblioteca, posteriormente los algoritmos diseñados para la aplicación específica y por último, la interfase con el usuario. Debido a la importancia de una estandarización en el proceso de adquisición, en esta sección se incluye como el cuarto punto la propuesta.

### **1. Biblioteca de funciones.**

Se consideró que si bien la mayoría de las veces las imágenes utilizadas serían de 256 niveles de gris, codificables dentro de una variable de un byte (unsigned char), sería conveniente contar con la posibilidad de

## IMAGEN

ima	col	ren	tam_var	tipo_var
Apuntador al primer elemento de la imagen. Sin tipo.	Número de columnas de la imagen. Variable tipo unsigned long.	Número de renglones de la imagen. Variable tipo unsigned long.	Tamaño de cada elemento de la imagen. Variable tipo unsigned.	Tipo de variable de la imagen particular. 2 Bytes.

**FIGURA 4.** Esquema de la estructura de datos "imagen", utilizada para manejar la información de la imagen.

manejar imágenes con otro tipo de variables (enteras de dos bytes, reales o complejas), por lo que se seleccionó una estructura de datos que permitiera esta posibilidad. A esta estructura se le llamó imagen (figura 4) y está formada por cinco campos:

- Apuntador a la dirección en memoria donde se encuentra la imagen.
- Número de columnas de la imagen.
- Número de renglones de la imagen.
- Tamaño de cada elemento de la imagen.
- Tipo de variable de los elementos de la imagen.

El apuntador a la dirección en memoria donde se encuentra la imagen no tiene especificado el tipo de variable al que apunta (void pointer); al momento de utilizar dicho apuntador se debe indicar de que tipo es la variable que direcciona por medio de un modificador de tipo.

La biblioteca está formada por varias unidades, que en general pueden dividirse en tres tipos: una general, las diseñadas para el manejo del sistema y las de procesamiento de imágenes.

En el anexo 1 se encuentra el código de los archivos de definición de los procedimientos programados.

**a). Unidad General.**

En esta unidad están las definiciones de la estructura de datos "imagen" y de la estructura de datos "curvas", con tres campos que corresponden a los valores en X, en Y y otro adicional, para el graficado y manejo de curvas (figura 5).

## CURVAS

<b>x</b>	<b>y</b>	<b>y1</b>	<b>numero</b>
Apuntador al arreglo de abscisas, tipo flotante.	Apuntador al arreglo de ordenadas, tipo flotante	Apuntador al arreglo auxiliar de ordenadas. Tipo flotante.	Número de datos.

**FIGURA 5.** Esquema de la estructura de datos "curva", utilizada para manejar la información de gráficas y curvas de datos.

Además en esta unidad se definen los parámetros del sistema (número de renglones y de columnas de la pantalla, niveles de gris, nivel del blanco, nivel del negro), macroinstrucciones para cada tipo de variable con el fin facilitar su utilización y las funciones generales que pueden ser utilizadas en cualquier otro procedimiento. Estas funciones son:

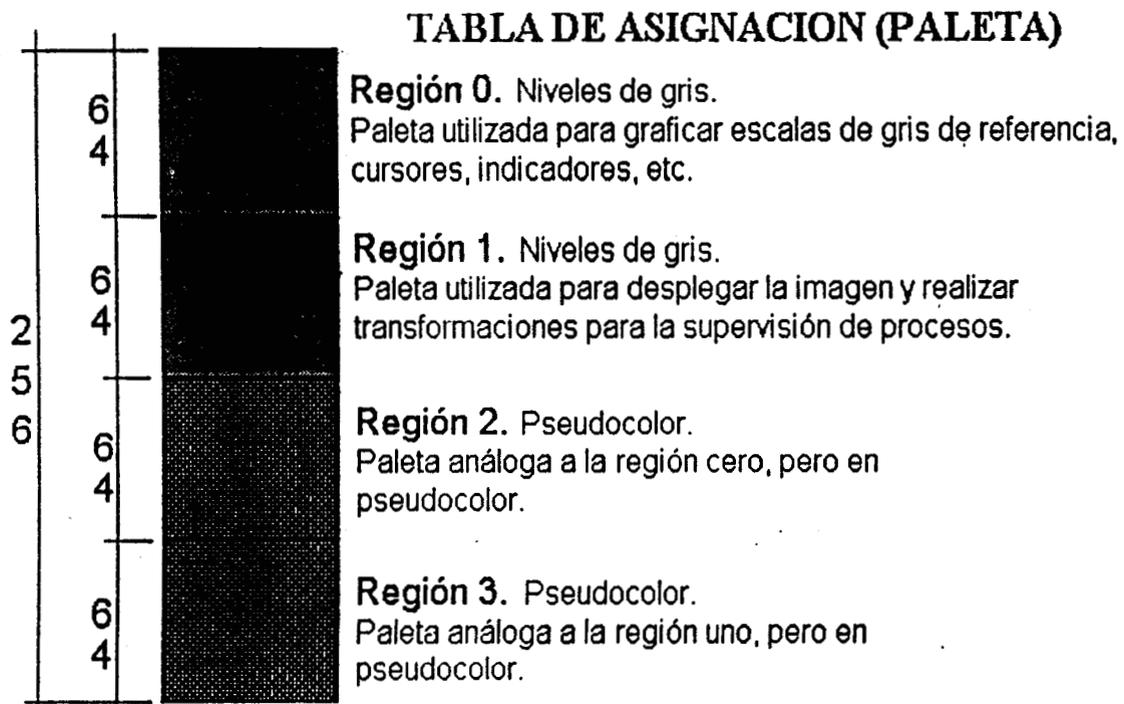
- espera\_tecla** : espera hasta que se oprime una tecla específica.
- round** : redondea el valor introducido como parámetro.
- ordena\_menor** : ordena dos enteros de menor a mayor.
- maxmin** : encuentra los niveles de gris máximo y mínimo de una imagen.
- max\_min\_datos** : encuentra los valores máximos y mínimos de cada campo de la estructura curvas.
- obten\_valor** : dado un apuntador sin tipo y el tipo de variable al cual apunta, se obtiene el valor de la variable direccionada por el apuntador.
- mete\_valor** : dado un apuntador sin tipo, el tipo de variable que se va a introducir y el valor de la variable, introduce dicho valor en esa dirección.

- apuntador** : dada la posición de un dato en coordenadas (x,y) dentro de la imagen, determina la dirección lineal del dato.
- beep** : da un sonido a cierta frecuencia durante cierto tiempo.

Dado que las dimensiones de la mayoría de las imágenes que se manejarán con el sistema (512x512), ocupan un espacio de memoria considerable, la estructura imagen sólo se define si el modelo de compilación utilizado es "large" o "huge".

**b). Unidades de manejo del sistema.**

**Sistema.** Las macroinstrucciones y funciones para el manejo del sistema de despliegue se definieron en esta unidad. El modo de video para el despliegue de imágenes seleccionado fue el modo 13 (320x200 pixeles con 64 niveles de gris).



**FIGURA 6.** Diagrama de la distribución de la tabla de asignación de niveles de gris para el despliegue de imágenes en modo 13.

Es importante destacar que si bien el despliegue se realiza sólo con los 64 niveles de gris permitidos por el sistema computacional, en resto de la programación se mantienen los 256 niveles de la imagen.

Para el despliegue de la imagen se definieron dos paletas, una de niveles de gris y otra correspondiente al pseudocolor. La tabla de asignación se dividió en cuatro regiones de 64 niveles cada una. En las dos primeras regiones se repitió la paleta de niveles de gris y en las dos últimas la de pseudocolor. Lo anterior se diseñó de esa manera para permitir el despliegue simultáneo en grises y color, además de permitir la modificación de una de las regiones de la tabla, manteniendo las demás inalteradas. En la figura 6 se muestra un diagrama de la forma en que se dividió la tabla de asignación.

En el caso de la paleta de niveles de gris, el nivel más bajo (cero) es el negro y el más alto (63 para despliegue, 255 real) es el blanco. La paleta de pseudocolor se definió con niveles crecientes de rojo, decrecientes de azul y crecientes hasta 32 y decrecientes en adelante para el verde, originando un nivel bajo color azul y un nivel alto rojo.

Las funciones realizadas en esta unidad son:

- detecta\_modo** : determina cual es el modo de video en el que se encuentra el sistema.
- cambia\_modo** : cambia el modo de video a un nuevo modo.
- paleta** : inicializa la tabla de asignación con las cuatro paletas ya descritas.
- pon\_paleta** : define una paleta y la da de alta en una de las regiones de la tabla de asignación.

**Memoria.** Dado que el tamaño de las imagenes es en general grande y que no es reservado durante la compilación, sino durante la ejecución del programa, se diseñó una unidad que manejara asignación y liberación de memoria para las imagenes y que apuntara a la memoria de video. Las funciones programadas son:

- mem\_video** : genera apuntador a la dirección de video (A000:0000).
- borra** : libera la memoria utilizada por una imagen.
- define\_imagen** : en base al número de renglones, de columnas, el tipo de variable y el tamaño de la variable en bytes, se reserva espacio en la memoria para una imagen; la salida es una estructura "imagen".

**Archivos.** Esta unidad se encarga del manejo de archivos en disco para la lectura y escritura de la imagen. Está diseñada de tal manera que indica la existencia de cualquier problema que pudiera sucitarse. El formato de imágenes que lee es de dos tipos: si el nombre no tiene extensión, supone que todo el archivo es la imagen, que el tipo de variable es de un byte (unsigned char), que la información está organizada por renglones y pregunta el número de renglones y de columnas; si el nombre tiene extensión ".ima", lee primeramente un encabezado que contiene el número de renglones, de columnas, el tipo de variable y el tamaño de ésta, posteriormente lee la imagen. Para la escritura a disco, el formato utilizado es el segundo. Las funciones accesibles a usuario correspondientes a esta unidad son:

lectura : lee imagen de disco. Regresa una estructura "imagen".

escritura : escribe imagen a disco.

abre\_archivo: pregunta el nombre del archivo y lo abre ya sea para lectura o escritura. En escritura determina el formato.

**Despliegue.** El despliegue de la imagen es una de las partes medulares del sistema ya que de él depende la monitorización de los procedimientos y la selección de las imágenes a procesar. Por otro lado, en ocasiones también es necesario desplegar gráficas de ciertos resultados. Esta unidad permite el despliegue gráfico tanto de imágenes (modo 13H) como de gráficas (modo 12H).

Para el despliegue de imágenes, se consideró la pantalla como otra imagen, por lo que se asignó la dirección a la memoria de video como el campo del apuntador a la imagen en la estructura "imagen", 200 como el número de renglones, 320 como el de columnas, caracter sin signo como el tipo de variable, con tamaño de un byte.

Dado que las imágenes pueden ser mayores a los 320x200 pixeles, se diseñaron rutinas de movimiento de la imagen en la pantalla. Además se consideró la necesidad de analizar sólo un segmento de la imagen total y la importancia de contar con cursores dentro de la imagen.

Como ya se señaló, la imagen se despliega en 64 niveles de gris, por lo que se divide cada valor entre cuatro solamente para el despliegue. Si bien se pierde resolución visual, fue necesario hacer este ajuste por la capacidad del sistema de video.

También se consideró la posibilidad de desplegar imágenes cuyos elementos fueran variables reales, enteros, etc. Para esto, se transformó la imagen original a una temporal con elementos de un byte.

Ya que el despliegue puede ser en niveles de gris o en pseudocolor, se agregó un parámetro a las rutinas de despliegue para indicar que región de la tabla de asignación se utilizaría. Para despliegue en niveles de gris se utilizó la región 1, para pseudocolor la región 3, para el cursor y otros indicadores la región 0 para pseudocolor y la 2 para niveles de gris (figura 6). Lo anterior se hizo con el fin de resaltar los indicadores y de evitar que en el caso de modificar la paleta de la imagen que se está desplegando, se modificara el fondo u otras regiones desplegadas con paletas diferentes.

Las gráficas pueden ser de barras o de puntos, con una o dos gráficas en la misma escala para el segundo caso. La estructura de datos empleada para las gráficas es la descrita en la unidad General, "curvas".

Las funciones accesibles a usuario, disponibles en esta unidad son:

- despliega : despliega imagen con elementos de un byte a partir de cierta dirección (elemento de la imagen) en determinada posición de la pantalla.
- grafica\_imagen : despliega imagen con elementos de cualquier tipo en cierta posición de la pantalla.
- despliega\_imagen : despliega imagen con elementos de un byte en cierta posición de la pantalla.
- dibuja\_cursor : dibuja el cursor en cierta posición de la imagen.
- mueve\_cursor : mueve el cursor en la imagen por medio del teclado.
- borra\_cursor : borra el cursor de la imagen desplegada.
- seccion : obtiene una sección de una imagen, por medio de un proceso interactivo con el usuario, a través de cursores.
- graficas : grafica los datos de la estructura curvas según un tipo de gráfica especificado.

El movimiento del cursor tanto en 'mueve\_cursor' como en 'sección', así como el desplazamiento de la imagen en la pantalla, se realiza por medio de las teclas >, <, ^ y v dependiendo la dirección del movimiento. El desplazamiento es de un pixel al oprimir la tecla sola, de cinco si se oprime simultáneamente con la tecla <CONTROL> para el cursor y de diez para el movimiento de la imagen.

Utilizando las mismas teclas para el desplazamiento de la imagen, pero simultáneamente con la tecla <ALT>, se despliega la frontera de la imagen en esa dirección.

Ya que se utilizan las mismas teclas para el cursor y para el desplazamiento de la imagen en la pantalla, para seleccionar una sección, o para realizar otras operaciones en las que se requiera de ambos movimientos, se debe oprimir <C> para mover el cursor y <S> para la imagen. La selección de la sección se realiza marcando dos esquinas contrapuestas con los cursores. Las esquinas no deben estar necesariamente en la misma pantalla de despliegue.

Cada vez que se realiza un despliegue de la imagen, se debe oprimir la tecla <ESC> para continuar con los procedimientos.

### **c). Unidades de procesamiento de la imagen.**

**Histograma.** Una de las transformaciones puntuales de realce de imágenes más comunmente empleada es la transformación del histograma de niveles de gris. Su extensa utilización se debe a su cualidad de manejo del contraste de la imagen, permitiendo mediante la modificación del histograma definir diferentes patrones de contraste para realzar ciertas características de la imagen (15).

Se generó una unidad que permitiera realizar diversas transformaciones del histograma incluyendo las de uso más común y permitiendo en un momento dado, la generación de transformaciones no definidas en la unidad.

En todas las transformaciones, se consideró como el rango del histograma de salida, el rango completo de las imágenes que se están manejando; es decir del nivel 0 al 255.

Las transformaciones más comunes son: la elongación lineal del histograma, para aumentar el rango dinámico de la imagen de trabajo; la elongación lineal pero negativa del histograma, generando un "negativo" de

la imagen con el histograma elongado; la igualación del histograma, tratando de equilibrar el número de pixeles con cada nivel de gris; y la especificación, en este caso de la transformación del histograma, mediante la cual el usuario puede definir como desea transformar los niveles de gris de su imagen original .

No se incluyó la especificación del histograma ya que se consideró de poca utilidad por su requerimiento de conocimientos sobre la representación en base a un histograma de la apariencia de la imagen.

Además de las transformaciones mencionadas, se incluyeron otras por considerarlas con posibilidades de aplicación. Dichas transformaciones fueron: exponencial, para elongar en mayor medida los niveles altos; logarítmica, para elongar principalmente los niveles bajos; senoidal (0 a  $\Pi$ ), niveles más altos de salida para los niveles centrales de entrada y más bajos de salida para los de entrada bajos o altos; senoidal (0 a  $\Pi/2$ ), mayor elongación en los niveles bajos y centrales; cosenoidal (0 a  $\Pi$ ), mayor elongación para los niveles centrales y altos, pero negativa; cosenoidal (0 a  $2\Pi$ ), niveles más bajos de salida para los niveles centrales de entrada y más altos de salida para los de entrada bajos o altos.

La transformación del histograma se realizó por tabla de asignación o mapa (26). Una vez determinanda la función de transformación, se recorrió la imagen; determinándose el nuevo nivel de gris de acuerdo con dicha función y asignándolo al pixel correspondiente.

Llamando  $r$  al nivel de salida, o al nivel de entrada,  $r_{min}$ ,  $r_{max}$ ,  $o_{min}$  y  $o_{max}$  a los niveles mínimo y máximo de salida y entrada respectivamente, las funciones de transformación programadas fueron:

Elongación lineal:

$$r = \frac{r_{max} - r_{min}}{o_{max} - o_{min}} (o - o_{min}) + r_{min}$$

Elongación lineal negativa:

$$r = \frac{r_{max} - r_{min}}{o_{max} - o_{min}} (o_{min} - o) + r_{min}$$

Elongación exponencial:

$$r = (r_{\min} + 1) \left( \frac{r_{\max} + 1}{r_{\min} + 1} \right)^a - 1$$

$$a = ((o - o_{\min}) / (O_{\max} - O_{\min}))$$

Elongación logarítmica:

$$r = \frac{r_{\max} - r_{\min}}{\log(O_{\max} - O_{\min} + 1)} \log((o - o_{\min} + 1)(O_{\max} - O_{\min} + 1))^a$$

$$a = r_{\min} / (r_{\max} - r_{\min})$$

Elongación senoidal (0 a  $\Pi$ ):

$$r = (r_{\max} - r_{\min}) \operatorname{sen} \left( (o - o_{\min}) \frac{\Pi}{O_{\max} - O_{\min}} \right) + r_{\min}$$

Elongación senoidal (0 a  $\Pi/2$ ):

$$r = (r_{\max} - r_{\min}) \operatorname{sen} \left( (o - o_{\min}) \frac{\Pi}{2(O_{\max} - O_{\min})} \right) + r_{\min}$$

Elongación cosenoidal (0 a  $\Pi$ ):

$$r = \frac{r_{\max} - r_{\min}}{2} \left( 1 + \cos \left( (o - o_{\min}) \frac{\Pi}{O_{\max} - O_{\min}} \right) \right)$$

Elongación cosenoidal (0 a  $2\Pi$ ):

$$r = \frac{r_{\max} - r_{\min}}{2} \left( 1 + \cos \left( (o - o_{\min}) \frac{2\Pi}{O_{\max} - O_{\min}} \right) \right)$$

Igualación del histograma (15):

$$r_i = \frac{Q - 1}{M N} \sum_{n=0}^i h_o(n)$$

donde  $r_i$  =  $i$ ésimo valor de salida;  $Q$  = número de niveles de salida;  $M, N$  = dimensiones de la imagen;  $h_o$  = histograma original de la imagen.

Para la especificación de la transformación se dibujaron los niveles de gris de salida contra los de entrada y por medio del movimiento del cursor se permite al usuario la definición de dicha transformación con segmentos de recta.

### ARR\_HISTO

histo	max	min
Apuntador al arreglo del histograma. Variable tipo unsigned long.	Valor máximo del histograma. Variable tipo entero.	Valor mínimo del histograma. Variable tipo entero.

FIGURA 7. Esquema de la estructura de datos "arr\_histo", utilizada para manejar la información de los histogramas de grises.

En esta unidad se definieron macroinstrucciones (constantes) para identificar cada una de las transformaciones programadas y la estructura arr\_histo que tiene un apuntador al arreglo del histograma, el máximo y el mínimo de dicho histograma (figura 7). Las funciones incluidas son:

- histograma\_g : determina el histograma de niveles de la imagen.
- dibuja\_histo\_g : dibuja el histograma en el modo 13.
- transf\_histo\_g : realiza la transformación del histograma. según el tipo de transformación especificado y con o sin cambio previo sólo en la paleta (para supervisión).
- transf\_lineal : elongación lineal, con pendiente positiva o negativa.
- transf\_log : elongación logarítmica.

<b>transf_exp</b>	: elongación exponencial.
<b>transf_senPi</b>	: elongación senoidal (0 a $\Pi$ ).
<b>transf_senPi</b>	: elongación senoidal (0 a $\Pi/2$ ).
<b>transf_cosPi</b>	: elongación cosenoidal (0 a $\Pi$ ).
<b>transf_cos2Pi</b>	: elongación cosenoidal (0 a $2\Pi$ ).
<b>transf_equ</b>	: igualación del histograma.
<b>transf_esp</b>	: especificación de la transformación del histograma.

A excepción de la especificación de la transformación del histograma, las demás transformaciones pueden realizarse sin entrar al modo del despliegue gráfico.

Si se realiza la transformación seleccionando el despliegue en niveles de gris y no en pseudocolor, y permitiendo la entrada al despliegue gráfico, la transformación se realiza inicialmente sólo sobre la paleta de grises, permitiendo al usuario decidir si la transformación es la adecuada para su aplicación. De ser así, la transformación se realiza sobre la imagen original, de otro modo dicha imagen se mantiene tal y como estaba al iniciar el proceso (figura 8).

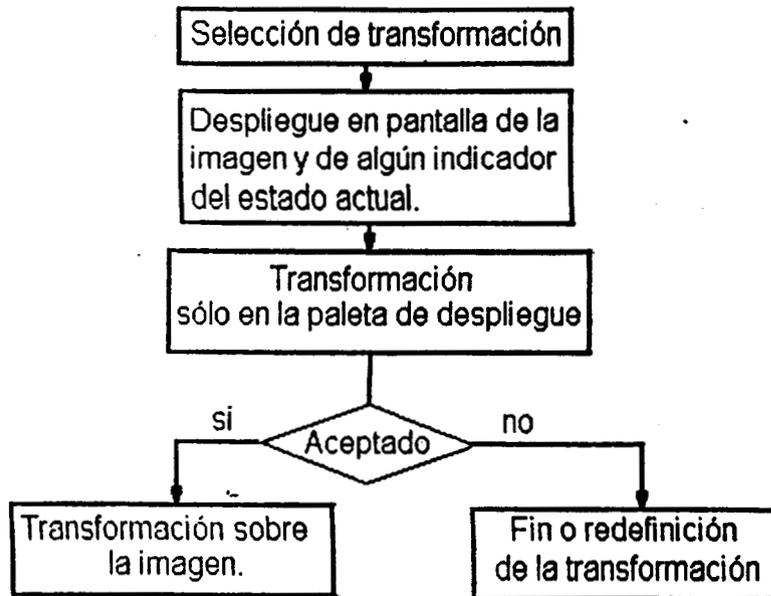
**Segmenta.** Muchas aplicaciones del procesamiento de imágenes requieren de una segmentación de la imagen, la mayoría de las veces por umbral; ésto es, se selecciona un valor de gris (umbral) y a los elementos de la imagen con niveles menores a él se les asigna un valor y a los de niveles mayores otro (15).

Se decidió incluir rutinas de segmentación de este tipo que contaran con dos umbrales, uno inferior y otro superior, para permitir la selección de un rango específico de niveles de gris. La segmentación se diseñó para ser realizada de cuatro maneras diferentes: asignando el nivel de blanco a los elementos con niveles entre los umbrales y negro a los externos; nivel negro a los internos y blanco a los externos; negro a los menores que el umbral inferior, blanco a los mayores que el umbral superior y se respetó el valor de los niveles intermedios; negro a los externos, manteniendo el valor de los niveles intermedios.

Al igual que en el caso de transformaciones del histograma, la segmentación puede tener una supervisión por parte de un operador, permitiéndole decidir los umbrales de segmentación a partir de cursores

colocados en una barra con todos los niveles de gris y modificando paralelamente la paleta de la imagen desplegada en la misma pantalla, de manera que pueda seleccionar interactivamente dichos umbrales. Si la segmentación obtenida no fue la deseada, el operador puede salir de la rutina de segmentación sin modificar la imagen de trabajo (figura 8).

### TRANSFORMACION CON SUPERVISION POR OPERADOR



**FIGURA 8.** Diagrama a bloques de las transformaciones supervisadas por operador.

La segmentación puede realizarse sin entrar al modo gráfico, definiendo los dos umbrales y el tipo de segmentación. Para el tipo de segmentación se especifican macroinstrucciones (constantes) en esta unidad. Las funciones accesibles a usuario correspondientes son:

- segmenta** : segmenta la imagen definiendo los dos umbrales y el tipo de segmentación.
- segmenta\_gama** : segmenta con la supervisión del usuario, con una gama de los niveles de gris de referencia, mediante el procedimiento descrito.

- segmenta\_p** : realiza la segmentación sólo en la imagen desplegada al modificar la paleta de despliegue.
- menu\_segmenta** : permite la selección de los tipos de segmentación en un menú dentro de un diseño para usuario con ventanas.

**Bordes.** El realce de bordes es una herramienta importante para delimitar objetos y mejorar la apariencia de ciertas imágenes. Si bien existen numerosos algoritmos para la detección de bordes (15,36), la mayoría se basa en realzar la señal espacial de alta frecuencia y por medio de umbrales determinar la región de borde.

De entre los numerosos operadores empleados para realzar los bordes, se seleccionaron los operadores diferenciales de gradiente (operador gradiente), dos aproximaciones a éste: la suma de los valores absolutos de las derivadas parciales en ambas direcciones y el máximo de las derivadas parciales en ambas direcciones, así como el laplaciano.

Las funciones de esta unidad son:

- gradiente** : operador de gradiente sobre la imagen.
- aprox\_max** : aproximación al gradiente por máximo.
- aprox\_sum** : aproximación al gradiente por suma.
- laplaciano** : operador de segundo orden laplaciano sobre la imagen.
- realce** : imagen menos laplaciano.

Cada que fue necesario, se realizó un escalamiento de la imagen para evitar que existieran valores negativos por la obtención de diferencias. Para evitar cálculos con elementos ya transformados, y dado que el resultado de los operadores se escribe sobre la misma imagen de entrada, la imagen de salida está defasada un pixel en ambas direcciones con respecto a la imagen original.

**Filtros.** Debido a la importancia de disminuir el ruido de ciertas imágenes o de resaltar características particulares, se incluyó una unidad con diversos procedimientos para el filtrado de imágenes.

Todos los filtros incluidos en esta unidad, se realizan en el dominio espacial. Se consideraron los filtros más utilizados (15) como son el de mediana para eliminar básicamente ruido impulsivo y el de promedio,

permitiendo la utilización de diversas ventanas de ponderación para generar filtros con características diferentes.

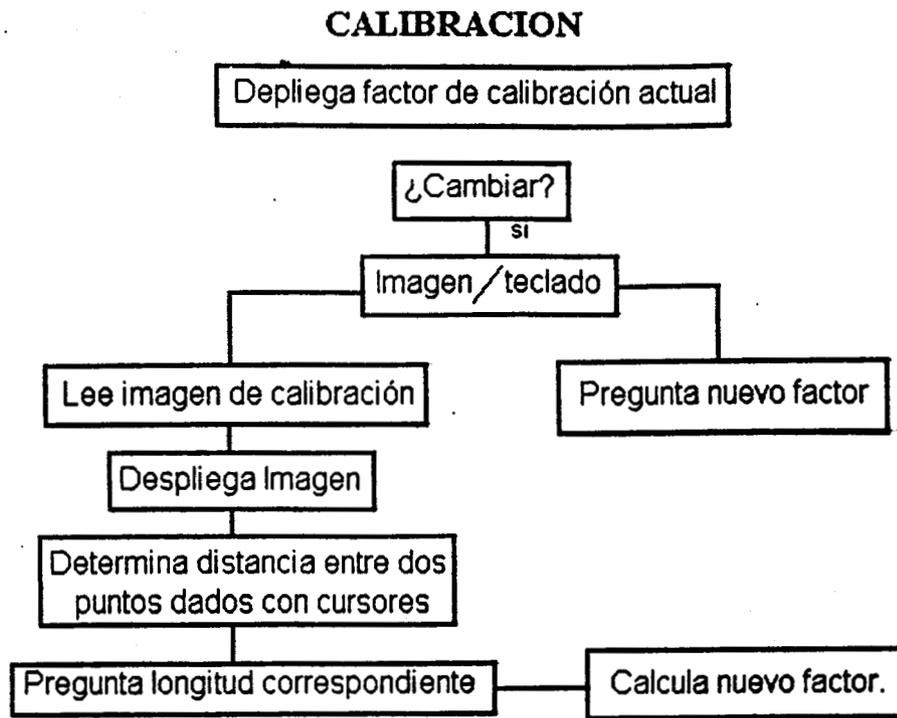
Los otros filtros incluidos (26) son: filtro sigma, con la posibilidad de modificar el número de desviaciones para realizar el filtrado; filtro de k próximos vecinos, con k seleccionable; y filtro de gradiente inverso. En todos los casos el tamaño de la ventana es seleccionable.

Las funciones utilizables por el usuario, definidas en esta unidad son:

- define\_ventana : pregunta los pesos que se utilizarán en el filtro promedio.
- filtro\_mediana : realiza el filtrado obteniendo la mediana de la ventana.
- filtro\_promedio : realiza el filtrado promediando los valores de la ventana, de acuerdo con una máscara de pesos para cada elemento de la ventana.
- filtro\_sigma : realiza el filtrado como el promedio de los valores de la ventana que entran dentro de cierto número de desviaciones estándar de la misma ventana en relación al pixel central.
- filtro\_kvecinos : realiza el filtrado como el promedio de los k vecinos más próximos al valor central de la ventana.
- filtro\_grad\_inv : realiza el filtrado en base al inverso del valor absoluto del gradiente de la ventana.
- filtrado : menú de selección de filtros dentro de un diseño para usuario con ventanas.

Al igual que en el caso de Bordes, el filtrado se realiza sobre la imagen de entrada y para evitar cálculos con elementos ya transformados, la imagen de salida está defasada el tamaño de la ventana entre dos con respecto a la imagen original.

**MediMorf.** Se consideró la necesidad de realizar mediciones de ciertas características geométricas sobre la imagen. Las características que se incluyeron fueron: longitud entre dos puntos, cálculo del área delimitando el diámetro de un círculo, cálculo del área de una elipse dando sus ejes mayor y menor, cálculo del volumen de un cilindro delimitando su diámetro y su longitud. Todos los parámetros necesarios para realizar los cálculos los delimita un operador mediante la utilización de un cursor sobre la imagen.



**FIGURA 9.** Diagrama a bloques del procedimiento realizado para determinar el factor de calibración.

En esta misma unidad se encuentran los procedimientos de calibración, la cual se realiza ya sea mediante la introducción de un factor de calibración ya conocido, o mediante la determinación en una imagen de una longitud a la que el operador asigna un valor (figura 9).

Las funciones de esta unidad son:

- largo** : determina la longitud entre dos puntos seleccionados por el operador.
- area\_st\_circulo** : determina el área de un círculo con diámetro igual a la distancia entre dos puntos seleccionados por el operador.
- area\_elipse** : determina el área de una elipse con diámetros seleccionados por el operador.

- volumen\_cilindro : determina el volumen de un cilindro con diámetro y largo seleccionados por el operador.
- mediciones : menú de selección de las mediciones mencionadas dentro de un diseño para usuario con ventanas.
- calibracion : determina el factor de calibración según el método ya descrito.

Todos los resultados de las mediciones se calculan de acuerdo con el factor de calibración presente en el momento de realizar la medición.

## **2. Determinación y análisis de la densidad anular de cultivos.**

Para determinar la densidad anular se divide al cultivo en regiones anulares del mismo ancho, definidas por círculos concéntricos trazados con el mismo cambio de radio, a partir del centro del cultivo. Posteriormente se calcula la densidad óptica media de cada anillo. Se asignó al nivel negro la máxima densidad y al blanco la mínima por considerar la zona de esporulación (negra), la de mayor biomasa. La densidad se calculó de manera relativa al valor máximo para procurar una estandarización.

Para iniciar el procesamiento de la imagen, es necesario someterla a una etapa de elongación lineal del histograma, con el fin que cualquier imagen correspondiente a un cultivo no se vea afectada por diferencias en el contraste. Se seleccionó una elongación lineal y no una igualación por la importancia que tiene para esta aplicación, la distribución de los niveles de gris. La elongación lineal no modifica dicha distribución, mientras que la igualación si lo haría.

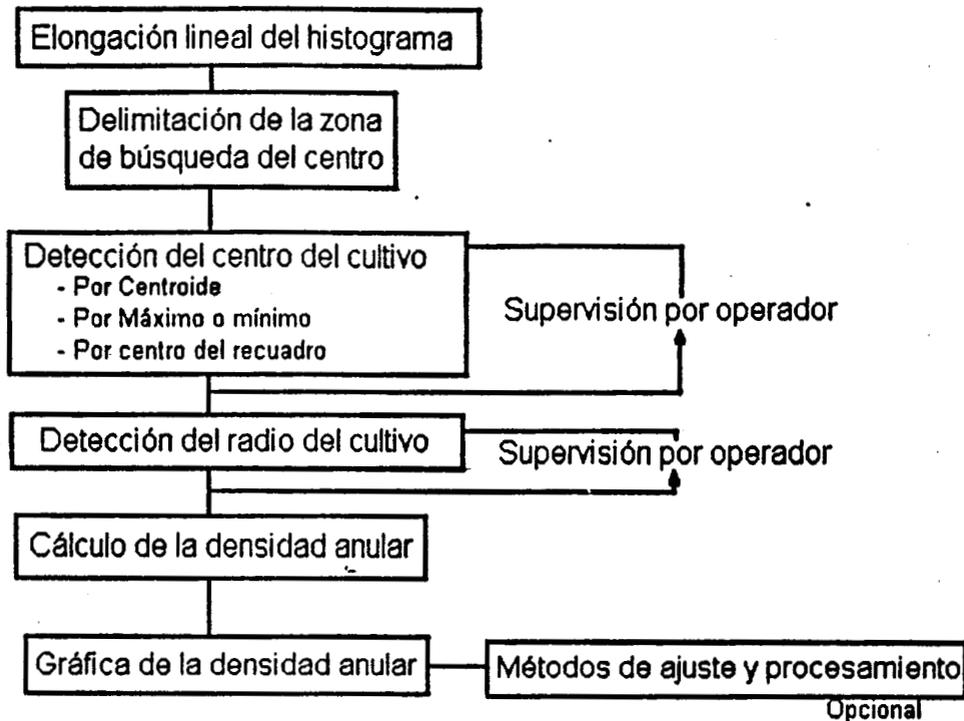
Una vez realizado este preprocesamiento de la imagen, la determinación de la densidad anular puede resumirse de la siguiente manera:

- Localización del centro del cultivo.
- Delimitación de los anillos.
- Determinación del radio total del cultivo.
- Determinación de la densidad de cada anillo.

En la figura 10 se resume en un diagrama el algoritmo para la determinación de la densidad anular de cultivos con crecimiento homogéneo. Para la descripción del procedimiento explicaremos inicialmente los algoritmos diseñados para la localización del centro del cultivo,

posteriormente la determinación del radio, el cálculo de la densidad anular y finalmente las unidades desarrolladas para realizar el procesamiento para el análisis de la densidad. En el anexo 2 se enlista la programación desarrollada para esta aplicación.

### DETERMINACION DE LA DENSIDAD ANULAR

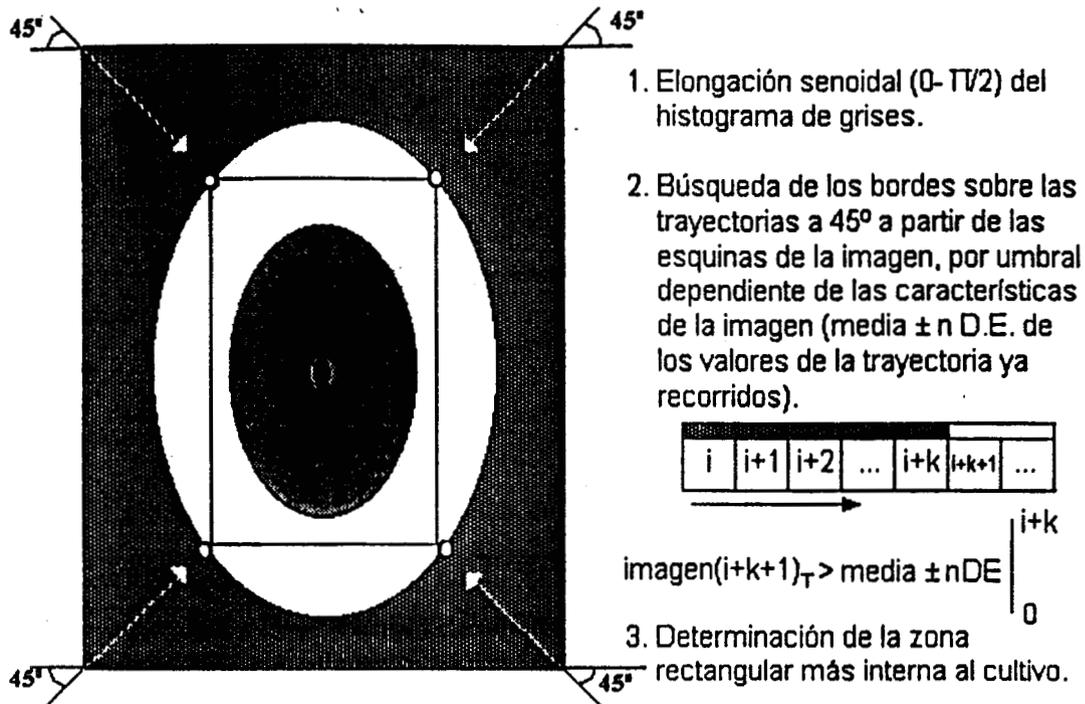


**FIGURA 10.** Diagrama a bloques de los procedimientos diseñados para la determinación de la densidad anular de cultivos con crecimiento homogéneo.

#### a). Localización del centro del cultivo.

El primer paso planteado en la localización del centro del cultivo fue la limitación de la zona de búsqueda, ya que en la detección del centro podía haber errores originados por objetos o zonas de la imagen que estuvieran fuera del cultivo y que afectaran los procedimientos de búsqueda del centro.

Delimitación de la zona de búsqueda del centro. La zona de búsqueda seleccionada fue una región rectangular interna al cultivo. El procedimiento para restringir esta zona se basa en la diferencia entre el nivel de gris del fondo (oscuro) y el del borde del cultivo (claro). En la figura 11 se esquematiza la delimitación de la región de búsqueda.



**FIGURA 11.** Diagrama esquemático de la determinación de la zona de búsqueda del centro del cultivo.  $n$  = número de desviaciones estándar a considerar al definir el umbral detección del borde; DE = desviación estándar de los valores de la imagen sobre la trayectoria; T = la trayectoria de búsqueda.

Dado que para la localización de la zona se requería realizar transformaciones sobre la imagen, se copió la información a una imagen temporal en memoria si había el espacio suficiente, o se guardó la información original en disco y se trabajó sobre la imagen en memoria. Se realizó una elongación senoidal ( $0-\pi/2$ ) del histograma de grises de la imagen para resaltar la diferencia entre el nivel del fondo y el del borde del cultivo. Esta transformación permite elongar el histograma casi linealmente

para los niveles bajos y centrales, disminuyendo la pendiente de elongación para los niveles altos. El nivel del borde queda casi blanco, mientras que el fondo permanece obscuro.

Para delimitar la zona de búsqueda del centro se detectaron los bordes del cultivo siguiendo una trayectoria diagonal a partir de las cuatro esquinas que delimitan la imagen. La detección de los bordes se hizo por umbral (36), pero por su característica de diferencia de intensidad en el borde ya no fue necesario utilizar algún operador para su realce.

Ya que la diferencia entre el nivel del fondo y el del borde puede variar entre las imágenes de los diferentes cultivos, principalmente por variaciones en el nivel del borde, se optó por un umbral dependiente de la imagen. Para determinar el umbral se fueron obteniendo la media y la desviación estándar de los niveles de gris de la trayectoria a 45 grados, comenzando desde la esquina. Los primeros cinco puntos de la trayectoria fueron utilizados para calcular los valores de la media y desviación estándar iniciales. Las ecuaciones empleadas fueron:

$$\text{media actual} = (\text{media anterior} (n - 1) + \text{valor actual}) / n$$

$$\text{desviación actual} = \sqrt{(\text{media actual}^2 - (\sum_{i=1}^{n-1} \text{valor}(i)^2 + \text{valor actual}^2) / n)}$$

donde n es el número de datos hasta el punto actual y valor es el nivel de gris de cada punto de la trayectoria.

Inicialmente, se consideró que existía un umbral si el nivel actual difería de la media por más de cinco desviaciones estándar. Sin embargo, para algunas imágenes el borde era más bien difuso, y la detección no la realizaba correctamente. En estos casos, se disminuía el número de desviaciones estándar hasta que la ordenada de los puntos localizados al iniciar en una esquina superior fuera mayor que al iniciar en una esquina inferior.

Como cabía la posibilidad de que los cuatro puntos localizados de esta manera no correspondieran a los vértices de un rectángulo, se tomaron los puntos más cercanos al centro del cultivo; esto es, se consideró como la zona de búsqueda al rectángulo más pequeño trazado con dos de los vértices opuestos localizados.

**Algoritmos de detección del centro.** Para detectar el centro del cultivo se hicieron las siguientes suposiciones:

1. Los cultivos a analizar crecen de manera homogénea, con simetría radial.
2. En los cultivos que presentaron esporulación, ésta comenzaba en el punto de inoculación (centro), por lo que esta zona presentaría mayor densidad.
3. En la mayoría de los cultivos sin esporular, la región central es más clara que el resto del cultivo (menor densidad).
4. Dado que el crecimiento del cultivo es en forma circular, su centro corresponde al centro de un rectángulo con los vértices en intersección con el borde del cultivo.

Los algoritmos diseñados para la determinación del centro del cultivo fueron:

**Por centroide.** En aquellos casos en los que el cultivo haya crecido de manera homogénea, es muy probable que el centro del cultivo corresponda al centroide de la región de búsqueda, considerando a la densidad como la función de peso. Las ecuaciones utilizadas para determinar el centroide fueron:

$$X = \frac{\sum_j \sum_i j (\text{nivel máximo} - \text{imagen}(i,j))}{\sum_j \sum_i (\text{nivel máximo} - \text{imagen}(i,j))}$$

$$Y = \frac{\sum_j \sum_i i (\text{nivel máximo} - \text{imagen}(i,j))}{\sum_j \sum_i (\text{nivel máximo} - \text{imagen}(i,j))}$$

donde (X,Y) corresponden a las coordenadas del centro del cultivo, nivel máximo al máximo nivel de gris (Blanco) e imagen (i,j) al nivel de gris de ese elemento de la imagen.

**Por detección de un máximo o un mínimo.** Considerando la característica de máxima o mínima densidad óptica del centro del cultivo, se propusieron tres procedimientos de detección. Dado que dependiendo del tipo de cultivo, la detección sería de un nivel de gris mínimo o de uno máximo, la primera parte de los tres procedimientos fue la determinación del tipo de cultivo. Se consideró como un cultivo con esporulación aquel cuyo nivel de gris del pixel localizado al centro del recuadro de búsqueda fuera menor que el nivel medio de la escala de grises (nivel  $< 128$ ) y sin esporulación aquel con un nivel de gris del pixel central del recuadro mayor o igual a dicho valor (nivel  $\geq 128$ ).

i) El primer procedimiento realiza una búsqueda del nivel mínimo o máximo de aquellos puntos en los que las pendientes de incremento o decremento del nivel de gris en la dirección X cambian; ésto es, que en tres puntos consecutivos en el mismo renglón de la imagen, la diferencia entre el primero y el segundo sea de signo contrario a la diferencia entre el segundo y el tercero.

ii) El segundo procedimiento tiene el mismo principio, sólo que la búsqueda se realiza tanto en la dirección X como en la Y, considerando como el centro aquel punto que resulte con el valor mínimo o máximo al recorrer la imagen en ambas direcciones.

iii) El tercer procedimiento utiliza una ventana de 3 x 3 pixeles. Se obtiene el valor medio de la ventana y se considera como el centro de la imagen al pixel central de aquella ventana cuyo valor medio sea el menor o mayor de toda la imagen. Este último procedimiento se diseñó pensando en evitar el problema de ruido en las imágenes que podía afectar a los otros dos algoritmos.

**Por centro de un rectángulo circunscrito en el cultivo.** El último procedimiento desarrollado para la determinación del centro del cultivo, se basó en que bajo las suposiciones ya descritas, el centro del cultivo debía corresponder al centro de un rectángulo circunscrito en la circunferencia correspondiente al borde del cultivo. Se consideró como dicho rectángulo al determinado para limitar la zona de búsqueda del centro.

Ya que el procedimiento completo se diseñó para ser realizado en forma semiautomática, con la supervisión de un operador, el centro determinado por cualquiera de los algoritmos descritos puede ser modificado por el usuario mediante un proceso interactivo. El centro se despliega sobre la imagen, permitiendo modificar su posición por teclado con las mismas teclas utilizadas para el movimiento del cursor. Mediante la tecla <ENTER> se selecciona el nuevo centro, tal como está desplegado o con <ESC> se acepta el centro determinado en manera automática.

**b). Determinación del radio del cultivo.**

El tamaño del cultivo se determinó como el promedio de la distancia de su centro a los vértices del recuadro de búsqueda. Se permitió la modificación de dicho tamaño por un operador mediante una rutina interactiva con el usuario. Sobre la imagen del cultivo se dibujaron los diámetros horizontal y vertical y se permite al operador aumentar o disminuir los radios hasta elegir el que le parezca óptimo. Para la modificación del radio del cultivo se utilizan las mismas teclas que en la modificación del centro. Para facilitar al usuario la visualización del proceso se marcaron de color rojo el cursor y el radio cuando éstos podían ser modificados y de azul cuando ya habían sido seleccionados.

**c). Calculo de la densidad anular.**

Una vez determinado el centro del cultivo, se procedió a calcular la densidad anular con respecto a dicho centro. La estructura utilizada para almacenar la densidad, "arr\_dens", consta de dos campos: el número de anillos y el apuntador al archivo con la información de cada anillo, desde el anillo central hasta el más externo (figura 12). La información de cada anillo se colocó en la estructura "densidad" con los siguientes campos:

- radio interno del anillo.
- radio externo del anillo.
- área del anillo (número total de pixeles \* factor de calibración de área).
- suma de las densidades (blanco - nivel de gris) de cada elemento del anillo.
- densidad del anillo (suma de las densidades / área del anillo).

## DENSIDAD

<b>r1</b>	<b>r2</b>	<b>area</b>	<b>nivel</b>	<b>dens</b>
Radio interno del anillo. Tipo de variable flotante.	Radio externo del anillo. Tipo de variable flotante.	Área total del anillo. Tipo de variable flotante.	Suma de niveles de gris de los píxeles del anillo. Tipo de variable flotante.	Densidad del anillo. Tipo de variable flotante.

---

## ARR\_DENS

<b>dens</b>	<b>num_anillos</b>
Apuntador al arreglo de densidad anular. Los elementos del arreglo son del tipo densidad.	Numero de anillos del arreglo Tipo de variable unsigned.

**FIGURA 12.** Esquema de las estructuras de datos "densidad" y "arr\_dens", utilizadas para almacenar y manejar la información de la densidad anular.

Para la determinación de la densidad se inicializó el arreglo con los valores de los radios, con una diferencia entre el radio externo y el interno de un píxel para obtener la máxima resolución; las demás variables se inicializaron en cero.

Para cada píxel de la imagen se calculó a que anillo correspondía, se incrementó el área de ese anillo y se sumó el valor de la densidad correspondiente a la suma total de densidades por anillo. Una vez recorrida toda la imagen, se calcularon las densidades de cada anillo. Considerando sólo los anillos internos al cultivo (radio externo < radio del cultivo), se normalizó la densidad anular dividiendo cada elemento del arreglo entre el valor correspondiente a la densidad máxima para normalizar los valores.

La densidad así determinada se despliega en una gráfica de barras y puede ser almacenada en disco con un encabezado correspondiente a una identificación del cultivo. El formato de escritura es en código ASCII,

importable a CUATRO, para permitir al usuario aplicar otro tipo de procedimientos sobre la información obtenida. Se almacena el radio externo de cada anillo y la densidad correspondiente.

**d). Unidades de procesamiento de la densidad anular.**

Se diseñaron dos unidades más para la aplicación, una para determinar la transformada de Fourier de señales unidimensionales y otra para realizar varios procedimientos de métodos numéricos. Estas dos unidades se generaron en forma separada al programa de determinación y análisis de la densidad anular por considerarlas de utilidad para otras aplicaciones.

Se incluyó la transformada de Fourier del arreglo de densidad ya que algunos cultivos presentan patrones anulares periódicos (figura \_) y se consideró que en esos casos la transformada de Fourier podría brindar información sobre ese aspecto. El procedimiento establecido permite determinar la transformada directa o inversa de Fourier por el algoritmo rápido (7), aproximando el número de puntos a la potencia de dos más cercana al número de píxeles del radio del cultivo. La transformación se despliega en una gráfica.

Los métodos numéricos se utilizaron para intentar ajustar a diversas curvas los valores de la densidad anular. Se implementaron los procedimientos necesarios para realizar un ajuste polinomial de grado  $n$  (9) y determinar el error del ajuste. Mediante el uso de esta misma rutina, se permitió realizar ajustes de otro tipo, exponenciales y a una gaussiana, ya que el ajuste puede ser en semilog o log-log.

Otro algoritmo incluido en la unidad de métodos numéricos fue la derivada de orden  $m$  de un arreglo (9). En el caso especial de la densidad anular esta determinación puede utilizarse para encontrar puntos de inflexión del arreglo con el fin de tratar de caracterizar su comportamiento, por lo que se incluyó en el programa de usuario la primera derivada del arreglo densidad.

Todos los procedimientos de métodos numéricos utilizan la estructura de datos "curvas" para realizar los cálculos. El primer y segundo campo del arreglo corresponden a los datos en X y en Y mientras que en el tercer campo, para el caso de los ajustes, se introdujeron los valores ajustados.

Tanto para el ajuste de datos como para la derivada de la señal, se graficó en pantalla el arreglo resultante.

Las funciones accesibles a usuario de estas dos unidades son:

Para la unidad de transformación de Fourier (FFT-1D):

- magnitud\_cuad : obtiene la magnitud cuadrada del arreglo.
- FFT : realiza la transformada rápida de Fourier en forma directa o inversa.
- grafica\_frec : grafica la transformación.

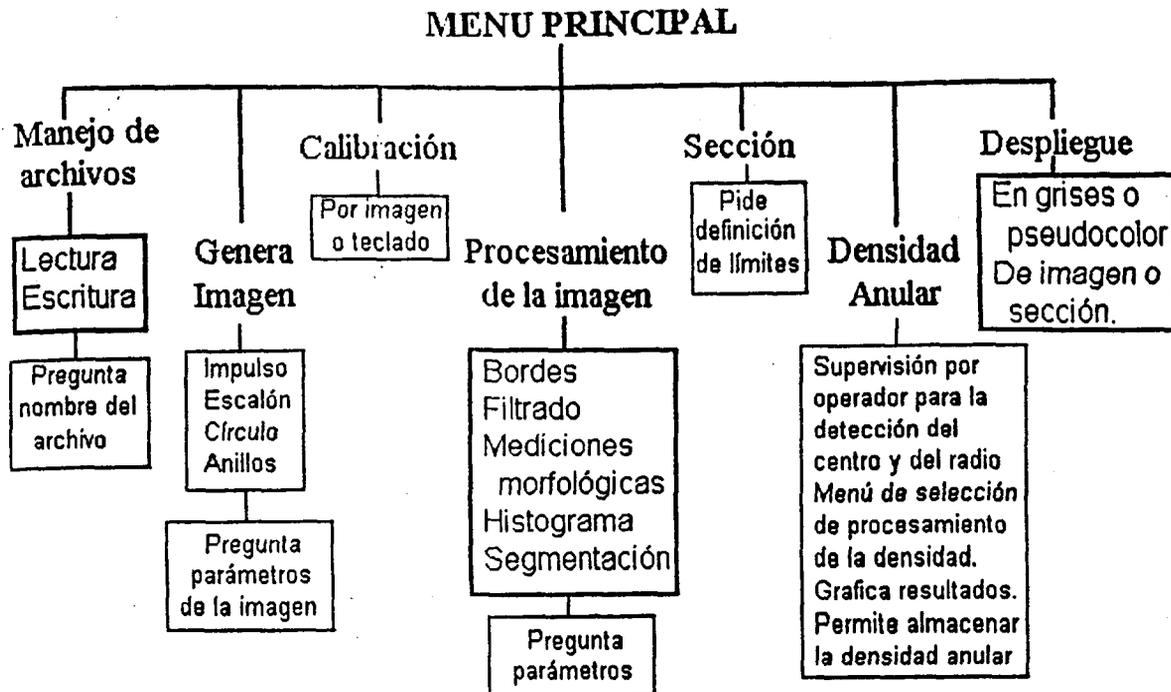
Para la unidad de métodos numéricos (MetNum):

- ajuste\_lineal : realiza el ajuste a una línea por mínimos cuadrados.
- ajuste\_polinomial : realiza un ajuste polinomial grado n y de tipo lineal, semilogarítmico o log-log de los datos en curvas. Regresa la matriz resultado con los coeficientes del polinomio.
- ajuste\_polinomio : realiza un ajuste polinomial con la rutina anterior, pero con determinación del error y con gráfica de los datos originales y los calculados con el polinomio.
- ajuste\_gausiano : realiza el ajuste a una curva gaussiana.
- ajuste\_exp : realiza el ajuste a una exponencial decreciente.
- derivada : obtiene la derivada orden m del arreglo.
- grafica\_ajuste : grafica el resultado del ajuste o de la derivada.

### **3. Programa de usuario.**

Se diseñó una interfase al usuario que le permitiera un manejo sencillo de los procedimientos incluidos. Se optó por una interfase con ventanas y menús de opciones para cada tipo de transformaciones o para la aplicación. Toda la interfase de usuario está en modo texto, salvo en el caso de despliegue de la imagen o de las gráficas de la densidad anular, de los ajustes, de la derivada del arreglo o de la transformada de Fourier.

Para aquellas rutinas que requieren de parámetros, se generan nuevas ventanas que preguntan por dichos valores. Cada vez que se está realizando un procedimiento cuya duración es perceptible para el usuario, se envía un mensaje sobre el proceso que se está realizando.



**FIGURA 13.** Diagrama de las opciones del menú principal y de las interacciones con el usuario el programa final. En letras grandes se muestran los grupos de funciones, en los recuadros con líneas gruesas las opciones completas y en los rectángulos de líneas delgadas, las opciones internas o los puntos de interacción.

En la figura 13 se presenta un esquema del menú principal, mostrando las opciones y los puntos de interacción con el usuario para cada grupo de funciones.

Como se puede observar, en el programa a usuario se incluyeron todos los procedimientos diseñados tanto para la aplicación específica, como para la manipulación y realce de la imagen.

Se definieron dos opciones de despliegue (Despliega y Muestra sección) ya que la primera corresponde a la imagen original, completa, mientras que la segunda muestra, si la hay, la sección de imagen obtenida mediante el procedimiento diseñado con este fin. Si se ha obtenido alguna sección, todas las transformaciones se hacen sobre dicha sección; en caso contrario, las transformaciones se realizan en la imagen completa.

La rutina de Generalm se diseñó para realizar pruebas con imágenes patrón. Está incluida en una unidad con sólo este fin y se buscó que las imágenes generadas cubrieran los requerimientos de los patrones de prueba (38). Las imágenes que se pueden generar son de tamaño variable (4x4 a 512x512) y pueden ser:

- Un impulso (fondo negro, impulso en blanco, con posición variable).
- Un escalón cuadrado (de tamaño y posición variable).
- Un escalón circular (de radio y posición variable).
- Un patrón de anillos concéntricos (posición del centro variable), con aumento exponencial del nivel de gris, según la función:

$$\text{imagen } [i,j] = \text{Blanco} \left( 1 - \exp \left( -10 r / (\text{número de columnas} \cdot \tau) \right) \right) \\ \left( 1 + \cos \left( 4 \Pi N r / (\text{número de columnas}) \right) \right)$$

con  $r = (i^2 + j^2)^{1/2}$ ; Blanco = nivel de gris del Blanco;  $\tau$  = razón de decaimiento de la exponencial; N = número de anillos.

En el último caso se pueden introducir alteraciones en los anillos, montando una senoidal sobre la circunferencia del anillo. También se le puede agregar ruido impulsivo negativo.

Se permitió una sola imagen fuente en la memoria, de manera que si ya existía una y se trataba de leer o de generar una nueva, el programa borraba automáticamente la primera antes de obtener la segunda. La imagen de la sección no cambia sino hasta elegir una nueva, en cuyo caso sucede lo mismo que con la imagen fuente.

#### **4. Estandarización del proceso de adquisición.**

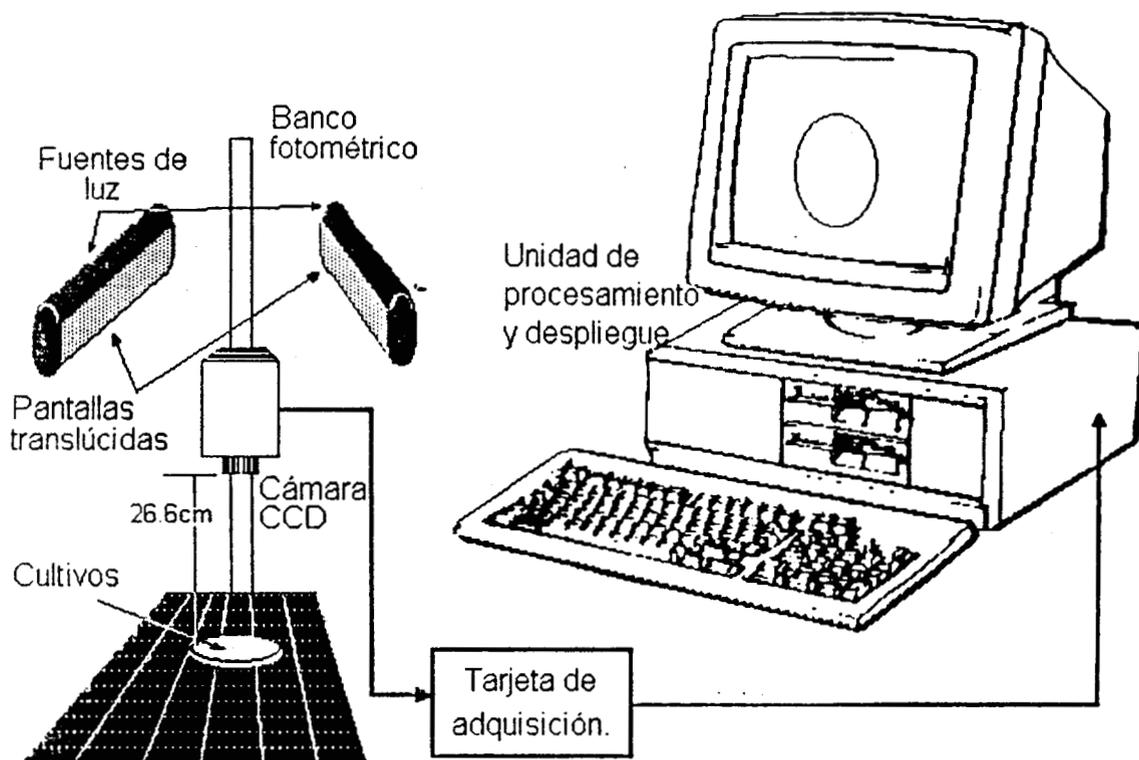
El proceso de adquisición es una parte medular para el correcto análisis de la densidad anular de cultivos. Las imágenes fueron adquiridas por medio de una cámara CCD EEV PS46510, se digitalizaron utilizando una tarjeta MATROX PIP1024, de 512x512 pixeles, con 256 niveles de gris y fueron almacenadas en disco para su procesamiento posterior.

Se utilizó como fuente de iluminación un banco fotométrico sobre el cual se colocaron los cultivos y se montó la cámara. El fondo del banco era negro mate, con una cuadrícula blanca de 3 cm<sup>2</sup> por cuadro. Dado que la iluminación resultó excesiva ya que las cajas de petri sobre las que se

encuentran los cultivos reflejaban la luz, se colocaron pantallas translúcidas, sobre cada una de las lámparas, logrando de esta manera, la iluminación adecuada. La luz no se dirigió directamente al cultivo.

La distancia entre la cámara y el cultivo utilizada para realizar la adquisición fue de 26.6 cm a la parte inferior de la caja de petri. Esta fue la mínima distancia que permitió la cámara, y se seleccionó por permitir una mayor resolución.

En la figura 14 se presenta un esquema del sistema de adquisición, procesamiento y análisis utilizado, incluyendo aquellos aspectos de la estandarización de la adquisición representables en un diagrama de este tipo.



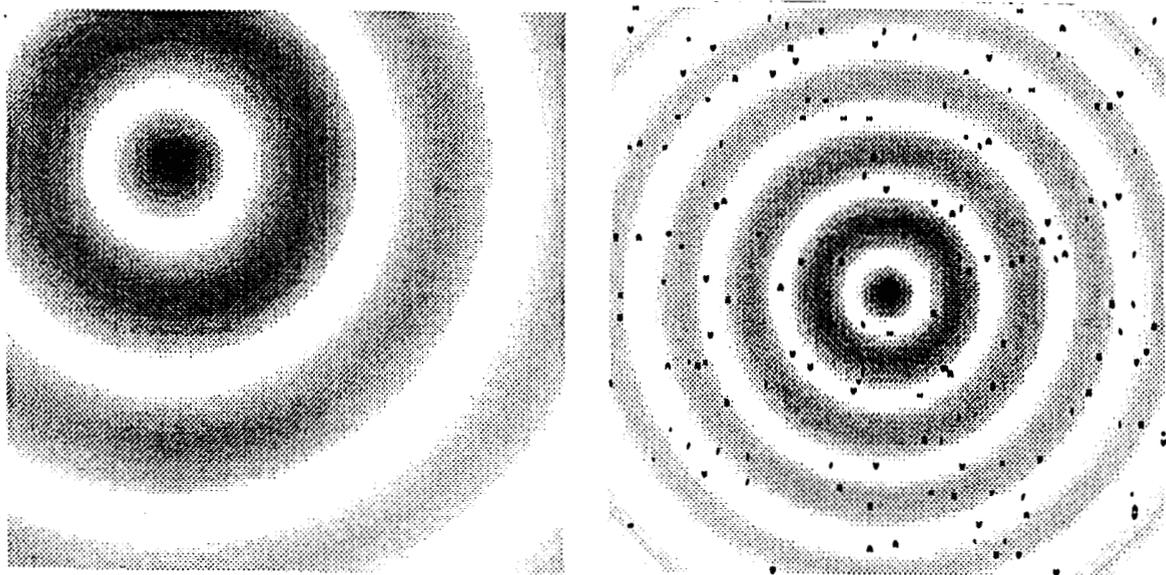
**FIGURA 14.** Diagrama general del sistema utilizado para la adquisición y análisis de la densidad anular de cultivos.

## Procedimientos de prueba del sistema.

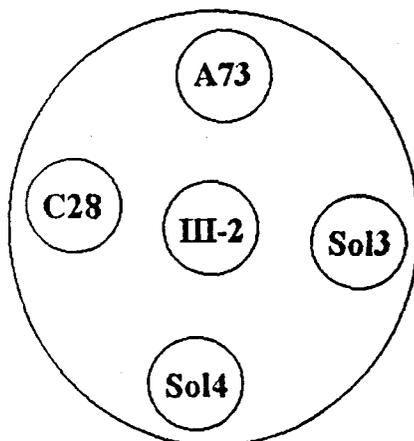
Las pruebas de funcionamiento de los procesos diseñados y de los programados con algoritmos ya definidos por otros autores, se realizaron en dos etapas: con imágenes patrón obtenidas con la rutina ya descrita para generar imágenes y con imágenes de cultivos adquiridas mediante una tarjeta de adquisición de imágenes (MATROX PIP1024).

Cada uno de los procedimientos generales de transformación de la imagen se probó durante su programación mediante imágenes generadas de diversos tamaños y características, para conocer el resultado de antemano y poder comparar con la imagen obtenida del sistema. Posteriormente se realizaron pruebas con imágenes adquiridas.

Para realizar las pruebas de los algoritmos diseñados para la determinación y el análisis de la densidad anular se utilizaron como imágenes patrón, la generadas con patrones de anillos concéntricos (figura 15). Cada algoritmo se probó con imágenes reales para realizar los ajustes necesarios y estandarizar el proceso de adquisición.



**FIGURA 15.** Ejemplos de imágenes patrón con anillos concéntricos. a) imagen sin ruido, con el centro recorrido hacia una esquina; b) imagen con ruido impulsivo, con el centros de los anillos y de la imagen iguales.



**FIGURA 16.** Ubicación aproximada de las mutaciones inoculadas en una misma caja de petri, vista desde arriba.

Una vez estandarizada la adquisición, se adquirieron las imágenes de 30 cultivos de *Aspergillus niger* obtenidos en el Area de Microbiología del Departamento de Biotecnología de la UAM-Iztapalapa de la siguiente manera: inoculación puntual de cinco mutaciones del hongo por caja; cada caja con un medio de cultivo con diferentes concentraciones de desoxiglucosa (0, 0.01, 0.05, 0.10, 0.30 y 0.50 gr/L).

Las mutaciones inoculadas fueron A73 (procedente de Holanda), Sol3, Sol4, C28 y III-2 obtenidas por exposición a la luz ultravioleta en la misma Area de la UAM-Iztapalapa. El lugar aproximado de inoculación en la caja de petri fue el que se muestra en la figura 16, viendo la caja desde arriba. En el momento de realizar la adquisición, los cultivos tenían una edad aproximada de una semana y no se observaba a simple vista interferencia entre los cultivos, a excepción de la caja con concentración de 0.30 gr/L de desoxiglucosa para el cultivo de la mutación Sol4, en la cual se podían observar dos puntos adicionales uno de los cuales interfería notablemente con dicho cultivo.

Dado que en cada imagen había más de un cultivo, se colocó en la región central de la imagen aquel que se deseaba analizar y con la rutina para obtener una sección de la imagen, se generaron imágenes que tuvieran sólo el cultivo de interés centrado en ellas.

La calibración se realizó mediante la adquisición de una imagen del fondo del banco fotométrico, para obtener el factor de calibración en base a la cuadrícula de .3 cm<sup>2</sup> ya mencionada. Se despreció la distancia correspondiente al grosor de la caja de petri por ser mucho menor a la distancia entre la cámara y el cultivo. Así mismo, a una distancia diferente, se adquirieron imágenes de una regla colocada de manera horizontal y vertical en el centro de la imagen. Con cada imagen se obtuvieron diez factores de calibración; se determinó la media, la desviación estándar y el coeficiente de variabilidad.

Para la calibración de las imágenes se utilizó el coeficiente obtenido con la cuadrícula del banco fotométrico, como el promedio de los diez factores obtenidos en diversas regiones de la imagen.

Para la verificación de los procedimientos de detección automática del centro y del radio del cultivo, se determinaron sobre la imagen desplegada en el monitor ambos parámetros de manera visual, posicionando un cursor sobre el punto que se consideraba el centro del cultivo y definiendo el radio aumentando o disminuyendo el tamaño del cursor. Este procedimiento se realizó cinco veces para cada cultivo.

Se compararon estadísticamente mediante análisis de varianza, las coordenadas del centro del cultivo obtenidas en las 30 imágenes analizadas, por los diversos algoritmos planteados y con el promedio de cinco mediciones del centro realizadas en forma visual. Esta última medición se consideró como el valor de referencia. Los otros criterios utilizados para la selección del mejor algoritmo de detección del centro del cultivo fueron el coeficiente de correlación entre los valores de referencia y los obtenidos por cada algoritmo y el porcentaje de error entre medias (valor absoluto de la media de los valores de referencia - la media de los valores obtenidos automáticamente entre la media de los valores de referencia).

Para la valoración de la determinación del radio del cultivo se utilizó la prueba de t para datos pareados, comparando el promedio de los cinco valores determinados de manera visual y el radio determinado de forma automática a partir del mismo centro del cultivo. Así mismo, se empleó el coeficiente de correlación entre ambas mediciones y el porcentaje de error entre medias.

Para todas las comparaciones estadísticas, el nivel de significancia utilizado fue de 0.05.

La unidad de métodos numéricos se probó con polinomios patrón y la de transformada de Fourier con señales con transformadas conocidas, tales como impulsos, funciones senoidales y escalones.

Estas dos unidades se utilizaron para realizar un análisis preliminar de la densidad anular de los 30 cultivos de *Aspergillus niger* adquiridos, con el fin de ejemplificar la caracterización. Las densidades anulares se ajustaron con un polinomio de orden 5, obteniéndose los coeficientes del polinomio y el error de ajuste. Así mismo, se ajustaron con la curva exponencial para delimitar el ajuste con menor error.

# RESULTADOS

Con el fin de facilitar el examen ordenado de los resultados obtenidos en las pruebas realizadas, se describirán dichos resultados para cada una de las partes que forman el sistema desarrollado.

## **Biblioteca de funciones y unidades de análisis.**

Tanto en las pruebas realizadas a los procedimientos generales de procesamiento de imágenes como a las unidades de métodos numéricos y de transformación de Fourier se pudo comprobar la correcta implantación de los algoritmos utilizados.

## **Calibración.**

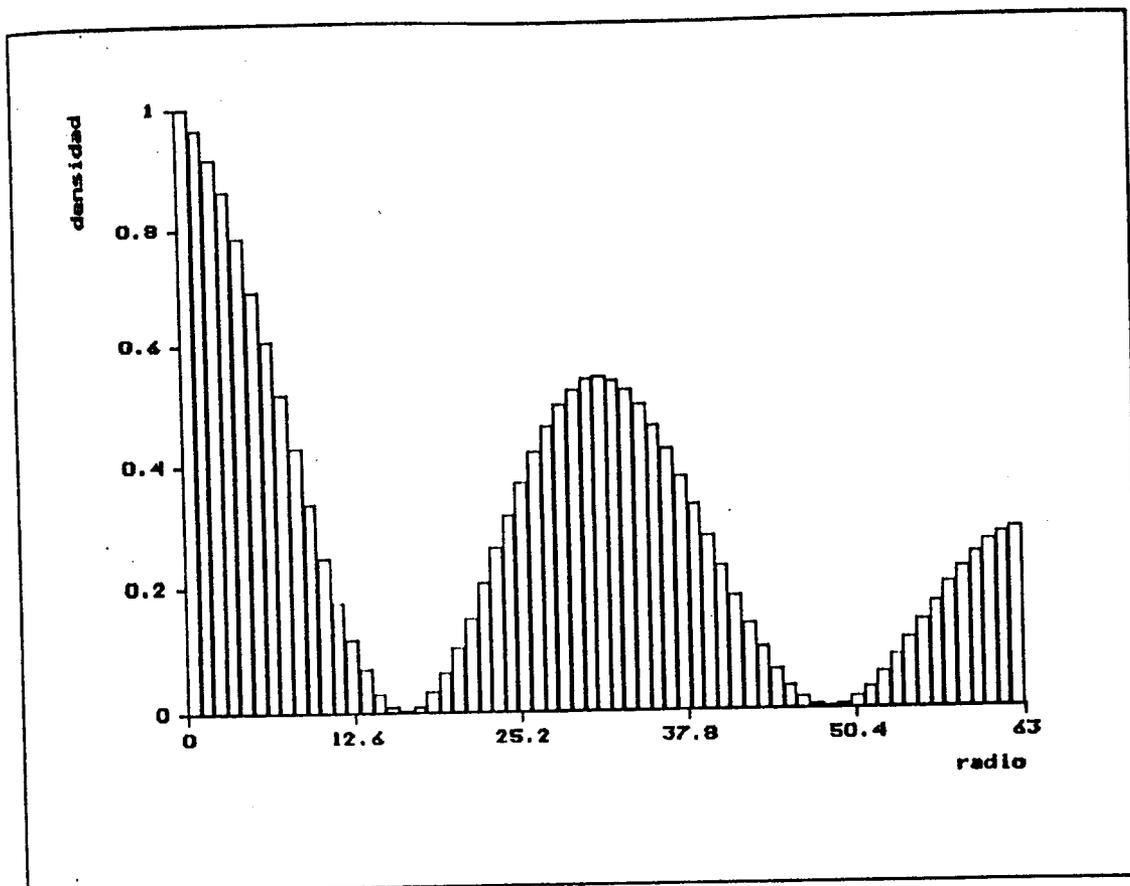
El factor de calibración obtenido con la retícula del banco fotométrico fue de 0.1221 mm/pixel con una desviación estándar de 0.0016 mm/pixel y un coeficiente de variabilidad de 1.3064 mm/pixel. Se analizó el error introducido por no considerar el grosor de la caja de petri y resultó ser del 1.1% para la imagen completa.

Los factores de calibración calculados con las imágenes de la regla en posiciones horizontal y vertical son diferentes estadísticamente. Para la imagen con la regla horizontal el factor de calibración fue de 0.1340 mm/pixel  $\pm$  0.7036 mm/pixel, con un coeficiente de variabilidad de 0.7036 mm/pixel, mientras que para la imagen con la regla vertical el factor calculado fue de 0.1112 mm/pixel  $\pm$  0.0010 mm/pixel, con un coeficiente de variabilidad de 0.9288.

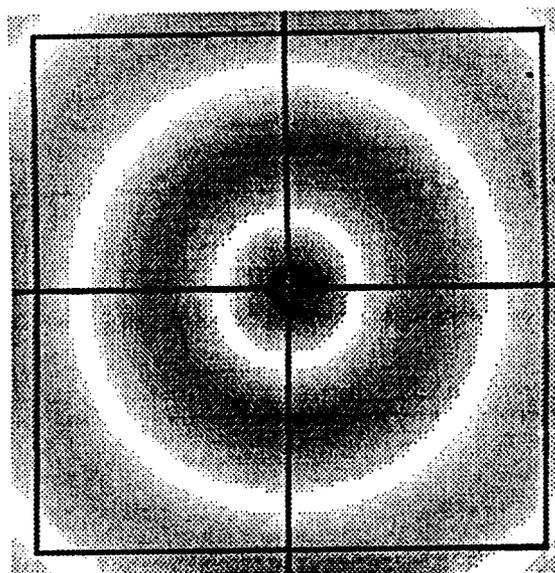
## **Densidad anular del cultivo.**

Las pruebas de la obtención de la densidad anular con las imágenes patrón se realizaron principalmente para probar las rutinas de detección del centro, utilizando toda la imagen como la región de búsqueda. Se observó que los algoritmos basados en la búsqueda de mínimos o máximos eran altamente susceptibles al ruido impulsivo, a excepción del algoritmo de media mínima. La determinación del centro por centroide resultó, para estas pruebas, ser variante al corrimiento. El algoritmo de detección por centro del

recuadro no se probó con este tipo de imágenes ya que el cultivo simulado no presentaba un borde con las características del real.



**FIGURA 17.** Ejemplo de imagen patrón de anillos concéntricos y la densidad anular determinada por el sistema.



**TABLA 1.** Coordenadas de los centros y radios determinados en forma visual y automática por los algoritmos planteados, para 30 cultivos de 5 cepas diferentes, cultivados en medios de cultivo con 6 concentraciones de desoxiglucosa.

Concentración gr/L	Cepa	X (píxeles)						Y (píxeles)						radio (mm)	
		v	c	mx	mm	mxy	r	v	c	mx	mm	mxy	r	v	a
0	A73	59.4	55	61	59	60	58	73.4	77	72	74	74	73	6.886	7.082
	SOL3	73.8	73	68	69	68	74	98.2	97	107	106	107	96	10.159	10.501
	SOL4	86.4	87	87	86	87	89	100.2	104	92	92	92	103	10.305	10.867
	C28	81.8	79	88	88	88	79	95.6	96	105	105	105	95	9.744	9.280
	III-2	69.2	68	44	31	44	69	87.0	89	96	31	96	89	8.400	8.547
0.01	A73	60.2	53	59	58	59	57	70.2	75	86	74	86	72	6.520	6.105
	SOL3	86.8	85	89	89	89	86	97.4	108	99	98	99	107	10.501	10.989
	SOL4	81.8	78	77	76	77	79	109.8	113	119	118	119	111	10.379	10.867
	C28	74.4	72	65	71	65	72	95.4	99	79	103	79	100	8.938	8.913
	III-2	65.6	65	64	64	64	65	83.2	86	82	82	82	85	7.863	7.814
0.05	A73	57.4	57	58	58	58	57	76.0	78	55	55	55	77	7.521	7.937
	SOL3	93.0	91	88	89	88	92	111.4	110	80	91	80	108	10.818	10.898
	SOL4	86.4	86	63	63	63	86	101.2	101	115	115	115	99	10.696	10.745
	C28	89.6	87	71	34	71	87	105.0	107	122	34	122	105	10.647	10.867
	III-2	84.8	80	86	86	86	80	97.6	99	118	117	118	101	9.719	9.646
0.10	A73	51.4	54	53	56	53	53	63.0	66	67	66	67	65	5.861	6.105
	SOL3	85.6	84	68	68	68	83	101.6	115	83	82	83	18	10.647	10.867
	SOL4	80.4	75	59	68	59	75	94.6	104	111	108	111	103	10.012	10.379
	C28	78.8	77	81	81	81	79	94.2	86	83	83	83	86	9.866	9.89
	III-2	69.2	69	69	70	69	69	79.6	80	81	80	81	78	8.156	7.937
0.30	A73	58.4	56	58	58	58	57	70.4	70	74	74	74	70	6.960	7.204
	SOL3	83.0	83	79	79	79	84	104.4	102	105	104	105	100	10.452	10.623
	SOL4	82.2	68	82	82	82	67	100.2	86	132	132	132	84	9.963	12.210
	C28	72.0	72	72	118	72	72	86.4	88	91	147	91	86	9.524	9.646
	III-2	75.0	73	87	87	87	73	87.0	86	118	117	118	86	8.816	8.791
0.50	A73	57.6	55	60	60	60	56	73.4	75	62	62	62	74	6.886	6.838
	SOL3	46.8	45	36	36	36	46	50.8	50	33	33	33	50	4.640	4.640
	SOL4	80.6	79	73	78	73	79	94.8	96	90	108	90	96	10.379	10.623
	C28	48.0	47	67	66	67	48	50.0	49	27	28	27	50	4.640	4.151
	III-2	79.4	78	86	86	86	79	92.4	94	90	90	90	94	9.304	8.669

v = método visual; c = por centroide; mx = por máximo o mínimo en dirección x; mm = por media mínima o máxima; mxy = por máximo o mínimo en ambas direcciones; r = por centro del rectángulo circunscrito; a = método automático.

La densidad anular obtenida en estas imágenes mostró que los procedimientos para calcular la densidad anular eran correctos. En la figura 17 se muestra una de las imágenes patrón utilizadas y la gráfica de su densidad anular.

Las pruebas parciales de la determinación de la densidad anular, sobre imágenes reales mostraron problemas en la delimitación de la región de búsqueda del centro del cultivo cuando las cajas de petri eran colocadas en superficies claras. También se observó que aparentemente no había ruido impulsivo en las imágenes adquiridas, por lo que no se planteó la utilización de filtros que lo eliminaran.

En la tabla 1 se encuentran los datos obtenidos para la validación de la detección del centro del cultivo y de su radio. La comparación por análisis de varianza de las coordenadas de los centros de los cultivos, obtenidas por los diferentes algoritmos planteados y la determinación visual (método de referencia) no mostró diferencias significativas ( $p < 0.05$ ). Los parámetros estadísticos así como los resultados de la comparación entre las coordenadas determinadas de manera visual y las de cada algoritmo de detección automática se muestran en la tabla 2.

**Tabla 2.** Características estadísticas de las coordenadas obtenidas con los diferentes métodos y comparación entre los valores determinados de manera automática y el método de referencia (visual).  $n = 30$ .

Coordenada	Método	media (pixel)	D.E. (pixel)	ERM %	r
<b>x</b>	visual	73.3	13.1		
	centroide	71.0	13.0	3.1	0.97
	min/max en x	69.9	13.7	4.6	0.72
	media mínima	70.5	18.4	3.9	0.47
	min/max en xy	69.9	13.8	4.6	0.72
	rectángulo	71.6	12.8	2.2	0.97
<b>y</b>	visual	88.1	16.1		
	centroide	89.5	16.9	1.6	0.96
	min/max en x	59.1	24.9	1.1	0.80
	media mínima	87.0	30.2	1.3	0.59
	min/max en xy	89.2	24.8	1.2	0.79
	rectángulo	88.1	16.5	0.2	0.96

D.E. = desviación estándar, ERM = error relativo de medias entre el valor de referencia (visual) y el método automático; r = coeficiente de correlación lineal entre el valor de referencia (método visual) y el método automático. min/max = método de detección del centro por búsqueda de valor mínimo o máximo.

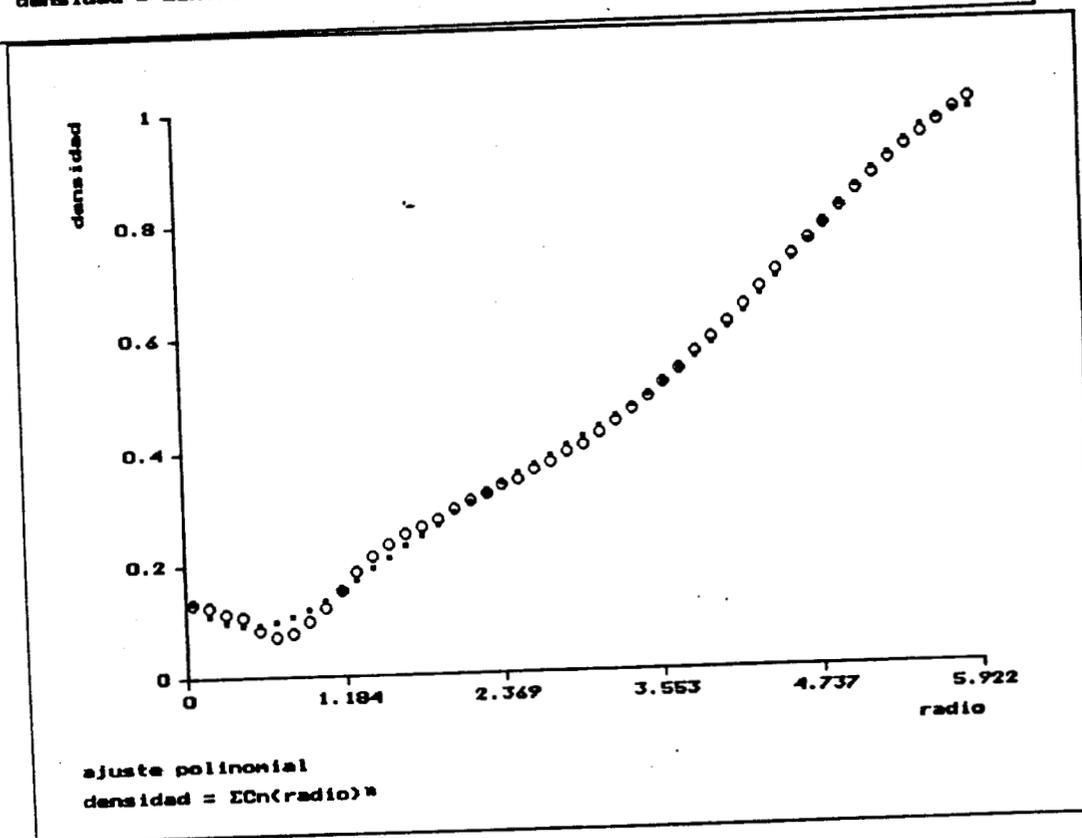
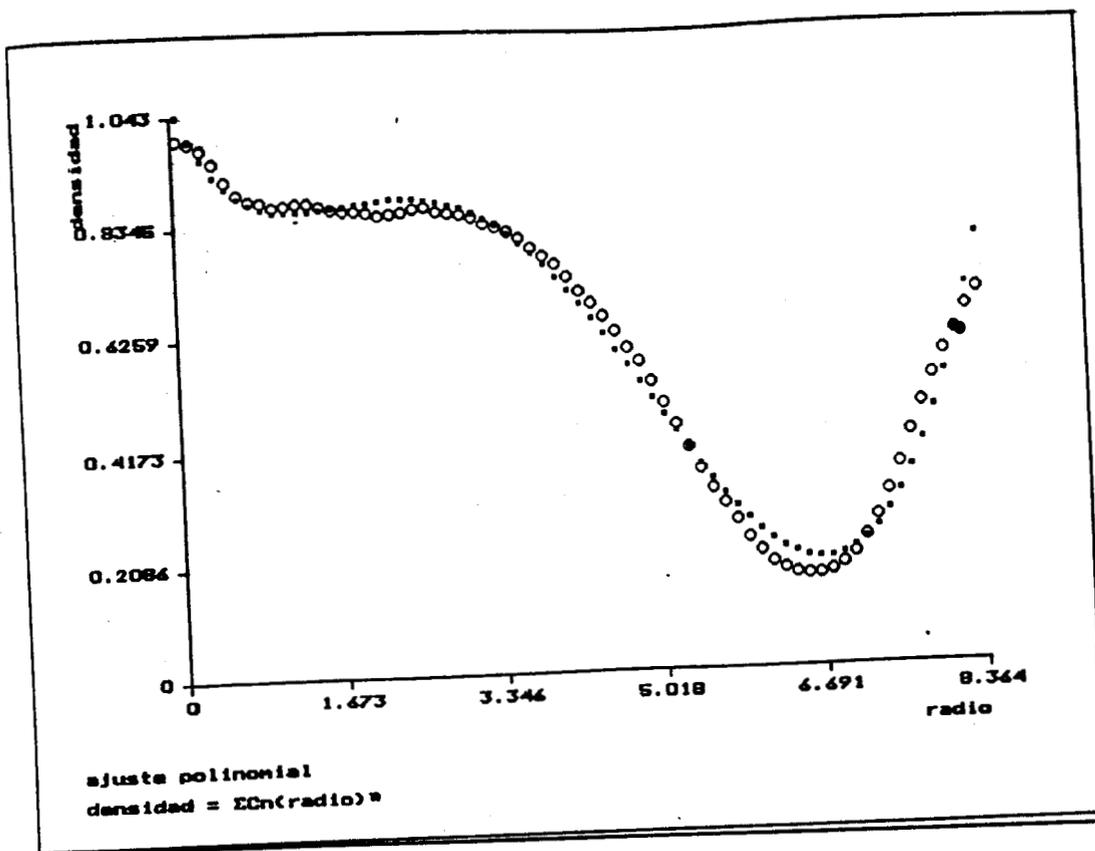


FIGURA 18. Gráfica de la densidad anular (círculos) y el ajuste con un polinomio de orden 5 (rectángulos), para un cultivo con (a) y uno sin esporulación (b).

El valor medio del radio determinado en forma visual (método de referencia) fue de  $8.84\text{mm} \pm 1.84\text{mm}$ , con un rango de 4.64mm a 10.82mm, mientras que el valor medio del radio determinado de manera automática fue de  $8.99\text{mm} \pm 2.04\text{mm}$ , con un rango de 4.15mm a 12.21mm. El error entre medias fue de 1.7% y el coeficiente de correlación entre ambas mediciones fue de 0.97. No se encontraron diferencias significativas entre los radios obtenidos de manera manual y automática por la prueba de t para muestras pareadas ( $p < 0.05$ ).

Los errores de ajuste con polinomios de orden 5 de los 30 cultivos analizados están entre 0.006 y 0.042 ( $0.022 \pm 0.009$ ). A manera de ejemplo, en la figura 18 se muestran las gráficas de la densidad anular de dos cultivos, incluyendo la gráfica del polinomio de ajuste.

Al tratar de ajustar la densidad anular con una exponencial, el error de aproximación siempre fue mayor que para el ajuste polinomial, con un rango de 0.024 a 0.248 ( $0.139 \pm 0.053$ ).

## ANALISIS Y DISCUSION DE RESULTADOS

Los resultados obtenidos durante las pruebas del sistema, demostraron que tanto la implantación de los algoritmos propuestos por otros autores, como de los diseñados en el presente trabajo tiene un funcionamiento adecuado. En forma particular, los procedimientos de determinación y análisis de la densidad anular de cultivos con crecimiento homogéneo mostraron una correcta operación para las imágenes reales que de éstos se obtuvieron.

Al igual que en la descripción del sistema y de los resultados obtenidos en las pruebas que se realizaron, se describirá el análisis de dichos resultados en forma separada para cada una de las partes que forman el sistema desarrollado.

**Biblioteca de funciones.** Contar con una biblioteca básica que definiera las estructuras y funciones generales para el procesamiento de imágenes facilitó el desarrollo de la aplicación específica diseñada para determinar la densidad anular de cultivos con crecimiento homogéneo, ya que además de formar una base sobre la cual se desarrolló el resto de la programación, permitió evaluar diversos procedimientos en una manera sencilla hasta encontrar el óptimo para la aplicación. Este fue el caso de la delimitación de la región de búsqueda del centro del cultivo. Dado que había que detectar los bordes del cultivo, inicialmente se utilizó una rutina de la unidad Bordes para tratar de localizarlo; sin embargo, como el fondo tenía cierta textura que resaltaba con los operadores diferenciales, hubiera sido necesaria la utilización de filtros para evitar detección en un lugar incorrecto, lo cual aumentaría el tiempo de procesamiento. Debido a que la diferencia entre el fondo y el borde del cultivo ya era notable en la imagen original, se trató de aprovechar esta característica, aumentándola mediante una transformación al histograma de grises; cambiando una única línea en la rutina de detección, se probaron varias transformaciones hasta encontrar aquella que resaltaba en mayor medida el borde del cultivo, siendo ésta la transformación senoidal (0 a  $\Pi/2$ ).

La biblioteca fue planeada en forma independiente a la aplicación específica, de manera que permitiera extender su utilización a otras

aplicaciones tanto en biotecnología como en otras ramas del conocimiento. Su correcto funcionamiento ya fue comprobado, y puede considerarse como la base tanto para las aplicaciones ya mencionadas como para la creación de una biblioteca más amplia de rutinas de procesamiento de imágenes.

En una de las aplicaciones del procesamiento digital de imágenes en biotecnología (20), los autores utilizan un sistema comercial que permite la programación específica con macroinstrucciones tipo Pascal proporcionadas por el mismo fabricante. La biblioteca propuesta se encuentra dentro de esta línea; en este caso no sólo consiste en un conjunto de instrucciones sino de una plataforma completa de programación utilizable únicamente bajo las mismas condiciones (lenguaje de programación, tipo de compilación, estructuras de datos, etc.).

**Estandarización de la adquisición.** Antes de analizar los resultados en la determinación de la densidad anular del cultivo, es importante comentar ciertos aspectos de la adquisición de las imágenes, incluyendo el proceso de calibración.

El diseño de un proceso estandarizado de adquisición y calibración permitió la comparación entre las imágenes de los diferentes cultivos analizados. Para lograr condiciones de adquisición equivalentes en cada caso de estudio se trataron de controlar la mayoría de las variables que afectaran dicho proceso. Entre esas variables se encuentran la iluminación, el fondo sobre el cual se colocó el cultivo, la distancia y relación geométrica entre la cámara y el cultivo así como el proceso de calibración.

La fuente de luz es un aspecto importante para la mayoría de los análisis de densidad óptica. Para lograr que la detección de los niveles de gris sea exacta y reproducible, es importante que la iluminación sea lo más constante y homogénea posible. El banco fotométrico permitió que se disminuyera la variabilidad de dicha fuente de luz y los filtros translúcidos utilizados eliminaron en buena medida los reflejos originados por los bordes de la caja de petri. A su vez, el fondo utilizado por ser color negro mate, no produce reflejos.

La cámara utilizada no fue la óptima para realizar un análisis macroscópico ya que por sus características de diseño no permitió un acercamiento suficiente para obtener la máxima resolución de la tarjeta de adquisición: la zona de interés de la imagen se encontraba en regiones

menores que 250x250 pixeles, disminuyendo de esta manera la resolución total a la cuarta parte o más. El hecho de montar la cámara sobre el banco fotométrico permitió estandarizar la distancia y posición de la misma, con el fin de mantener las condiciones de adquisición durante todo el periodo de estudio.

La cuadrícula blanca del banco fotométrico no interfiere con la delimitación de la densidad anular del cultivo ya que esta medición se está realizando sólo para la región interna donde por la opacidad del cultivo, las líneas no son observables. Algunos cultivos pueden presentar un borde difuso de baja opacidad, pero aún en estos casos el área de esta región es grande en relación con la ocupada por las líneas de la cuadrícula y la diferencia entre los niveles de las líneas y el borde no es grande, por lo que el error introducido es mínimo.

Un problema mayor es el originado por la diferencia entre los valores de calibración cuando las mediciones se hicieron en las direcciones vertical y horizontal. Este problema no está originado por la razón de aspecto del modo de despliegue utilizado (modo 13), que no es 1 a 1, ya que el factor de calibración está definido como el número de pixeles correspondientes a cierta longitud entre la misma longitud. Aún más, al observar una retícula fina a través del sistema de video utilizado con la tarjeta MATROX (con razón de aspecto 1:1), se observó que existía una transformación geométrica de la imagen adquirida. Probablemente este problema es originado por la cámara que se utilizó; el sistema de medición alteró la información de entrada, produciendo una transformación no deseada de la misma. La solución a este problema requiere de la caracterización de dicha transformación y la realización de un método del que se obtenga la transformación inversa (13,24). Este procedimiento queda fuera de los objetivos de la presente tesis.

Con el fin de minimizar los efectos de este problema, en el presente estudio se utilizó como factor de calibración para el análisis de los 30 cultivos, el promedio de diez factores obtenidos en diferentes regiones de la imagen de la base del banco fotométrico y la posición del cultivo en relación con el sistema de adquisición fue siempre el mismo (al centro de la zona de adquisición). En tanto no se solucione el problema planteado, se sugiere al proceso descrito como el estándar de calibración y adquisición.

**Densidad anular del cultivo.** En general, para todas las pruebas realizadas, la determinación de la densidad radial del cultivo se realizó de manera satisfactoria. Los perfiles de densidad así obtenidos permitieron distinguir características morfológicas de los cultivos a nivel macroscópico, tales como el radio del cultivo y la distribución con relación a su centro de las densidades ópticas del mismo.

Cuando se utilizaron las imágenes patrón ya descritas, los algoritmos para la determinación del centro del cultivo tuvieron un comportamiento adecuado. Sólo en la determinación por centroide se observó una dependencia con la posición del cultivo; si el centro del cultivo simulado se encontraba desviado hacia alguno de los lados, la detección resultaba incorrecta. Este problema resulta obvio cuando se considera que este algoritmo está afectado por la distribución de los niveles de gris dentro de la imagen. En imágenes cuya distribución de niveles no es igual en cada dirección, el centro detectado por este método se desvía hacia el lado que presenta valores de peso mayores es decir, hacia las zonas más oscuras.

Cuando se realizaron las pruebas preliminares con imágenes reales, sólo se utilizó la región de la imagen interna al cultivo determinada de la manera ya descrita. Dentro del rectángulo de búsqueda la distribución de los niveles está dada sólo por la densidad óptica del cultivo. Por las características de los cultivos estudiados (crecimiento homogéneo), esta densidad está distribuida en la mayoría de los cultivos en forma casi simétrica en relación con el centro del cultivo. En estas imágenes la detección del centro del cultivo por centroide no presentó los problemas descritos para las imágenes patrón, tal como era de esperarse.

Además de solucionar el problema descrito, la limitación de la zona de búsqueda evitó la detección errónea del centro del cultivo por los algoritmos de búsqueda de máximo o mínimo en aquellos casos en que otras zonas de la imagen, externas al cultivo, cumplían los requisitos de detección.

Sin embargo, al limitar la zona de búsqueda todos los algoritmos presentaron cierta dependencia con la posición del cultivo en la imagen completa. Esta dependencia la origina la forma en la que se delimita la zona; en realidad el algoritmo variante al corrimiento es el que define el rectángulo

de búsqueda. Este problema no es importante en la detección del centro para cualquiera de los algoritmos mientras se cumplan los siguientes puntos:

- a). el cultivo esté totalmente incluido en la imagen,
- b). la trayectoria de búsqueda de los bordes pase por la circunferencia del cultivo en los cuatro cuadrantes.

Como el usuario puede seleccionar una sección de la imagen sobre la cual se realice el procesamiento, mediante ese procedimiento se pueden cumplir ambos requisitos.

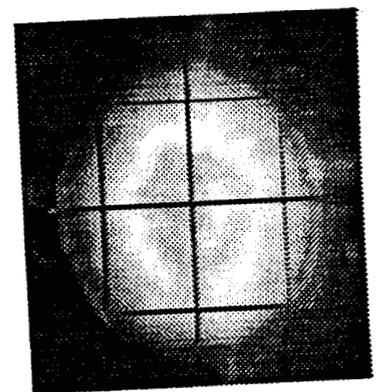
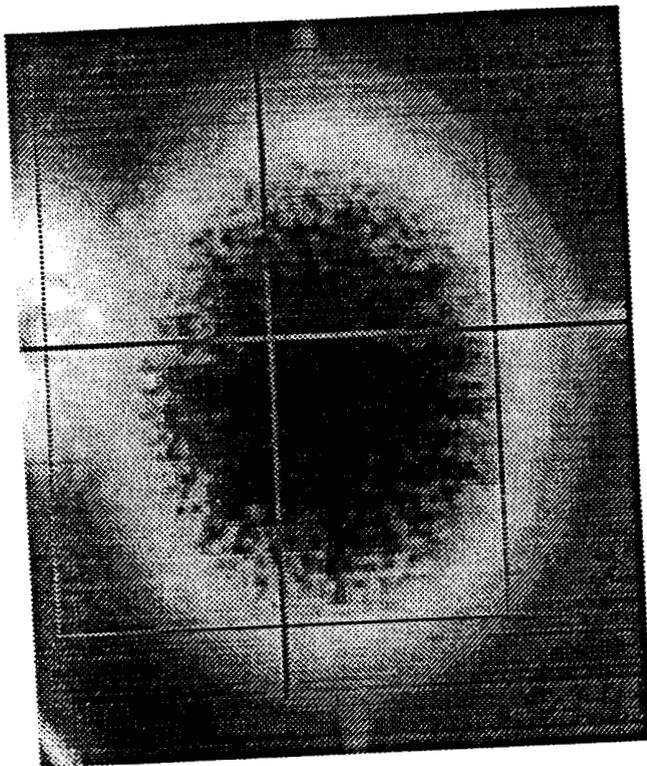
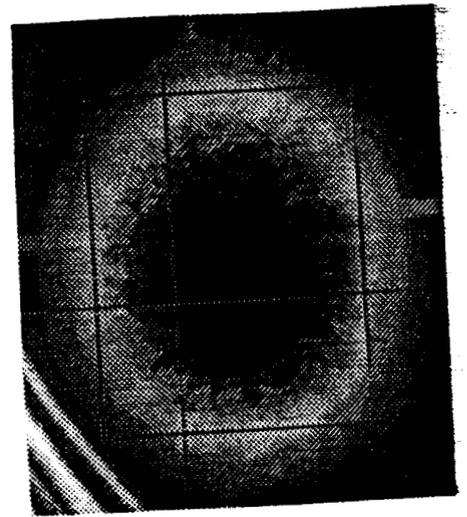
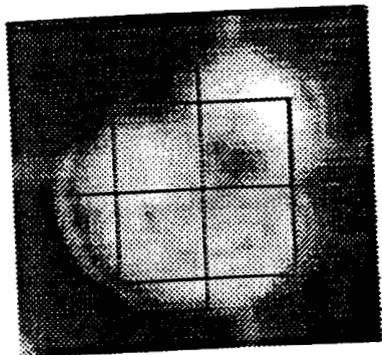
Durante la prueba del sistema se observó que cuando el cultivo se inoculó cerca de la pared de la caja de petri, ésta podía interferir en la detección de los bordes del cultivo. Este problema puede evitarse si la adquisición de la imagen se realiza antes de que el cultivo alcance la pared o si los puntos de inoculación se eligen de manera que el cultivo no llegue hasta ese punto. Por otro lado, si el cultivo alcanzara la pared de la caja de petri, esto interferiría con su desarrollo produciendo errores metodológicos más graves, por lo que se debe evitar totalmente esta condición.

Otro problema que puede generar una delimitación errónea de la zona de búsqueda es el originado por cultivos con crecimiento no homogéneo (figura 19.a), ya que en estos casos, el recuadro puede no estar centrado en la imagen. Sin embargo, la determinación de la densidad anular para este tipo de cultivos podría cuestionarse por no presentar simetría radial.

En el cultivo con interferencia de otro punto de inoculación accidental (Sol4 para 0.30 gr/L de desoxiglucosa), la determinación de la zona de búsqueda fue errónea (figura 19.b). En la tabla 1 se puede observar que para ese cultivo, tanto las coordenadas del centro como el radio determinados automáticamente, son las que presentan mayor variación con respecto a la determinación manual. Generalmente, este tipo de cultivos no son considerados por el biotecnólogo para el análisis de los cultivos por que generan resultados erróneos que pueden desviar la información correcta.

Los principales problemas para la detección del centro del cultivo con los diferentes algoritmos, observados en los 30 cultivos analizados fueron:

- 1) Zonas más oscuras que el centro en otra región del cultivo.
- 2) Clasificación como cultivo esporulado o no esporulado errónea.
- 3) Problemas en la delimitación de la zona de búsqueda.



**FIGURA 19.** Principales problemas observados en la determinación del centro y el radio de los cultivos analizados. a) crecimiento no homogéneo; b) interferencia con otro cultivo; c) regiones con nivel más bajo que el centro en otra zona del cultivo; d) clasificación errónea del tipo de cultivo (esporulado/no esporulado).

El problema 1 afectó a los algoritmos de localización del centro por búsqueda de mínimo para los cultivos con esporulación (figura 19.c). El problema análogo no se presentó en los cultivos sin esporulación ya que en todos los casos de este tipo, la región más clara fue el centro.

El segundo problema se presentó en un sólo caso (figura 19.d) donde existía esporulación, pero el nivel de gris de esa zona era menor 128, por lo que se consideró al cultivo, para la detección de su centro, como no esporulado. El problema se manifestó en los algoritmos de determinación por mínimo o máximo.

Los problemas en la determinación de la zona de búsqueda ya se describieron y afectaron principalmente al último algoritmo por su dependencia directa con el rectángulo que define la zona; en ocasiones el algoritmo de detección por centroide también se vio afectado por generar un corrimiento para la búsqueda similar al que se presentaba en las imágenes patrón ya descrito.

A pesar de todos los problemas mencionados, no se observaron diferencias estadísticas por análisis de varianza entre las coordenadas obtenidas automáticamente por los diferentes algoritmos y las obtenidas visualmente. ( $p < 0.05$ ). Por esta razón, se utilizaron los criterios adicionales de coeficiente de correlación y el error absoluto entre las medias para las coordenadas visuales y las determinadas automáticamente (tabla 2) con el fin de seleccionar al algoritmo con el mejor comportamiento. Se consideró como el mejor algoritmo aquel que tuviera un coeficiente de correlación más alto con un error menor. Para los algoritmos probados, el de selección fue el del centro del recuadro de búsqueda, si bien los resultados de la detección por centroide fueron también satisfactorios. Por otro lado, el algoritmo de detección seleccionado es el más sencillo y el que consume menor tiempo de cómputo. Los tres algoritmos que detectan el valor máximo o mínimo son los más complejos y tuvieron un desempeño pobre, principalmente en lo relacionado con coeficiente de correlación. Este parámetro de comparación es muy importante ya que no sólo permite una comparación entre las medias de los valores, sino de la relación que existe entre ellos.

En la tabla 1 se puede observar que los valores obtenidos por los algoritmos de búsqueda por mínimo o máximo en la dirección X y el que

realizaba la búsqueda en ambas direcciones fueron iguales en todos los cultivos analizados, exceptuando A73 con 0.01 gr/L de concentración de desoxiglucosa. Lo anterior da una idea de que para cultivos con crecimiento homogéneo (simetría radial), existe invarianza a la rotación para la determinación del centro.

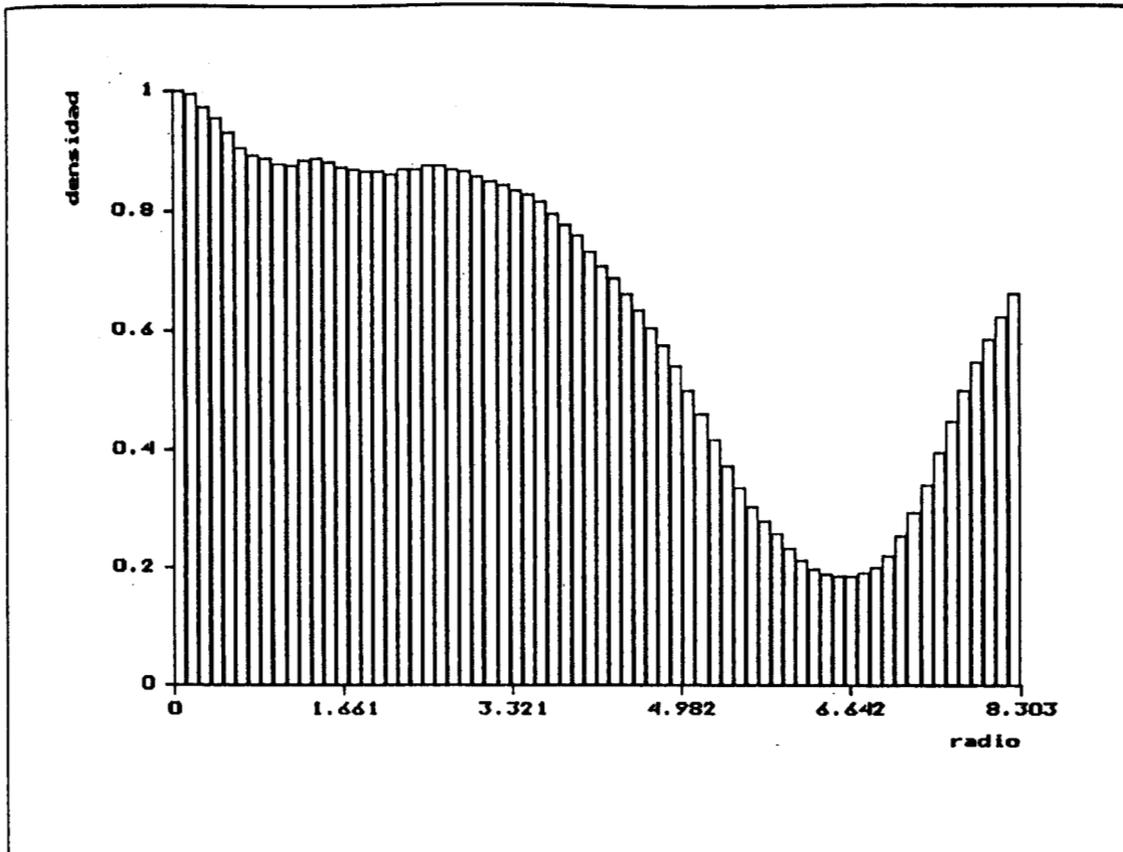
Dado que para la determinación del radio del cultivo la comparación estadística entre los valores radio determinados visualmente y los obtenidos de manera automática para el mismo centro, no mostró diferencias significativas ( $p < 0.05$ ) y la correlación lineal entre ambos métodos fue alta (0.97), con un error relativo entre medias de 1.7%, puede considerarse al método propuesto como una alternativa confiable.

Las diferencias observadas entre los dos métodos son debidas en la mayoría de los casos a alguno de los siguientes problemas:

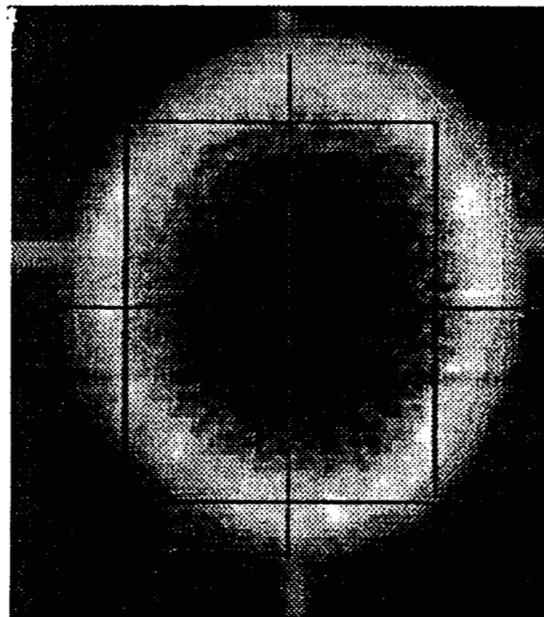
a). Como el radio del cultivo se calculó mediante el promedio de la distancia entre el centro y los vértices del recuadro de búsqueda del mismo, cualquier problema en la delimitación de dicho recuadro generó problemas en la delimitación automática del radio del cultivo.

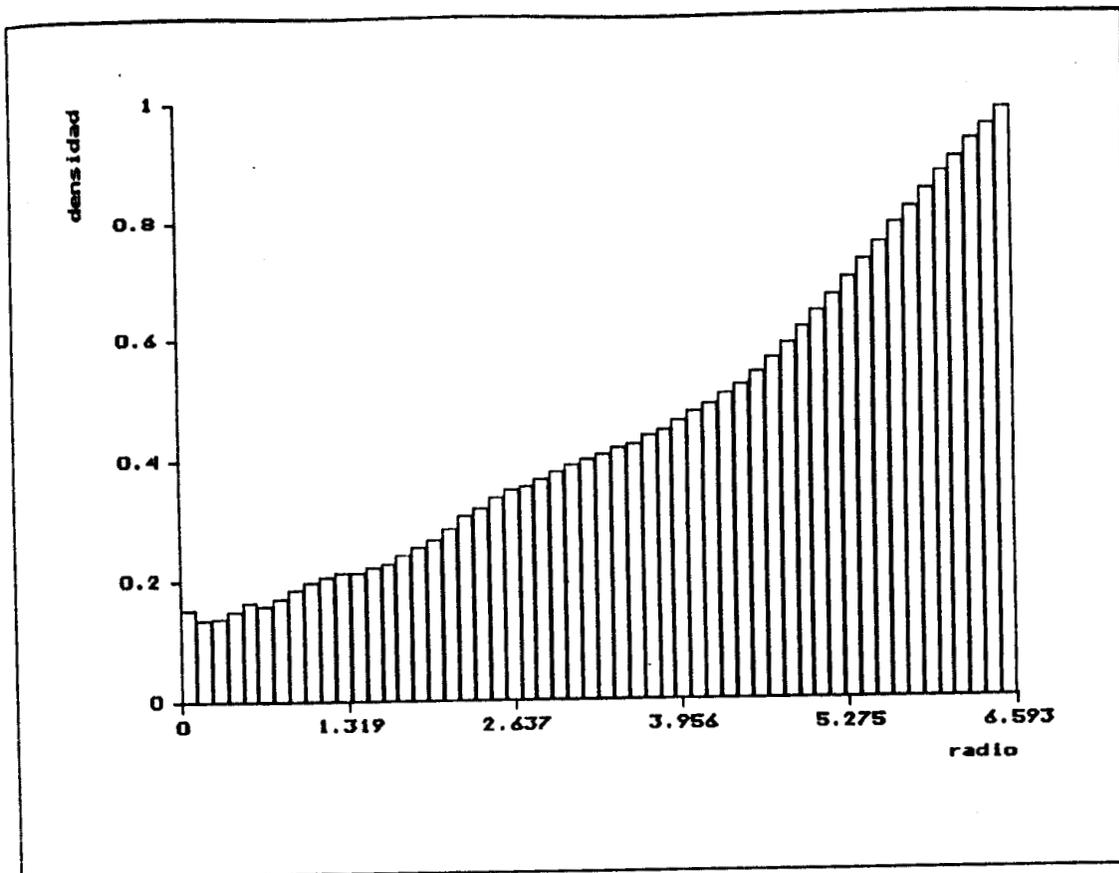
b). La transformación geométrica de la imagen ya descrita (sección de estandarización de la adquisición en este mismo capítulo) originó que prácticamente en todas las imágenes, el radio horizontal fuera menor que el vertical y el trazado a los vértices del rectángulo de búsqueda fuera de un valor intermedio. Al igual que en el caso de la calibración, se sugiere que si se determina manualmente el radio, se seleccione un valor intermedio entre los radios horizontal y vertical, en tanto se soluciona el problema de la transformación geométrica.

La detección con los algoritmos planteados de los parámetros morfológicos (radio y centro) presentó un desempeño adecuado, con los problemas ya descritos. Se pueden utilizar por tanto para una determinación totalmente automática de la densidad anular del cultivo; sin embargo, como pueden proporcionar resultados con los que el usuario no esté totalmente de acuerdo, es importante recalcar que en esta fase del proceso se permite al operador una edición manual es decir, la reubicación del centro y la redefinición del radio del cultivo.

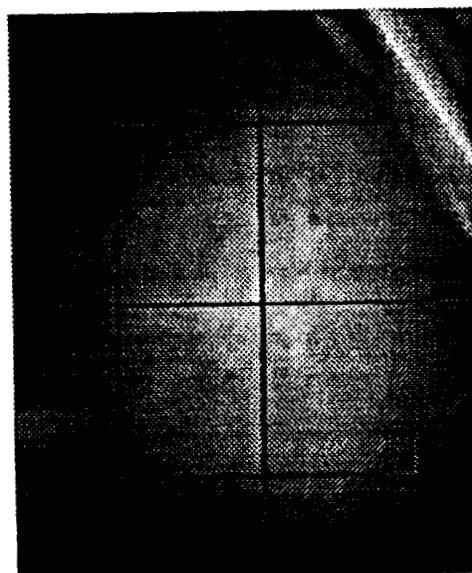


**FIGURA 20.** Gráfica de la densidad anular de un cultivo con esporulación (III-2 con 0.10 gr/L de desoxiglucosa) y la imagen correspondiente con el recuadro de búsqueda, el centro y el radio determinados por el sistema.





**FIGURA 21.** Gráfica de la densidad anular de un cultivo sin esporulación (A73 con 0.10 gr/L de desoxiglucosa) y la imagen correspondiente con el recuadro de búsqueda, el centro y el radio determinados por el sistema.



Las curvas de la densidad anular presentaron dos morfologías predominantes:

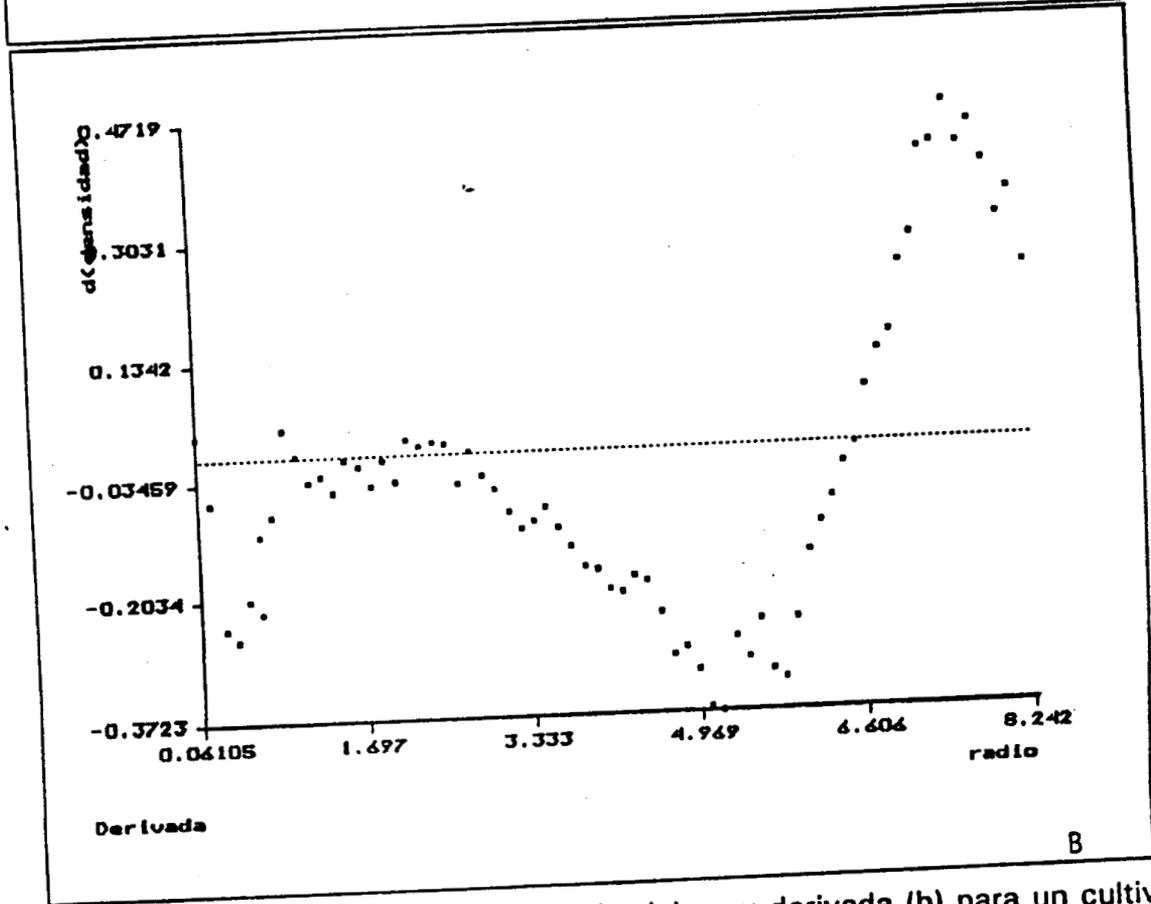
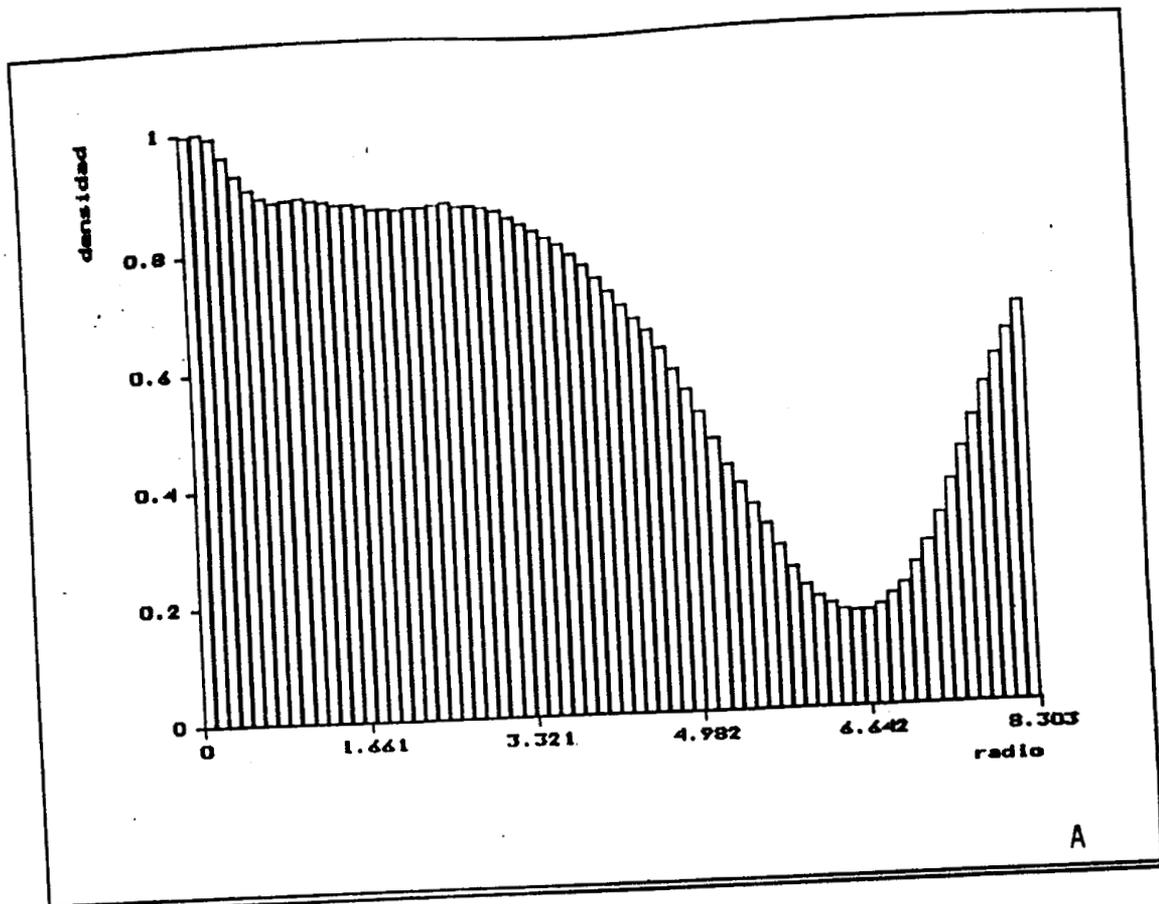
- Cuando el cultivo presentaba esporulación (figura 20), la curva era decreciente para toda la zona en la que el cultivo presentó alta opacidad, y creciente en la región externa del cultivo en la que la opacidad disminuye. Se observó una especie de meseta para los radios pequeños, en algunos casos antecedida por una disminución de la densidad.
- En el caso de los cultivos sin esporulación, la densidad anular fue creciente en relación al radio (figura 21), observándose en ocasiones una meseta para los anillos con radios mayores.

Los métodos numéricos incluidos en el programa están encaminados principalmente al ajuste de curvas por medio de polinomios. Se permitió la utilización de ajustes semilogarítmicos o log-log para aumentar el campo de aplicación de las rutinas incluidas. La otra función importante de la unidad MetNum es la obtención de la derivada del arreglo en tratamiento. Para todos los procedimientos programados se obtuvo un comportamiento satisfactorio al probarlos con funciones conocidas.

El ajuste con una curva exponencial fue pobre, presentando errores mayores a los obtenidos con el polinomio de orden 5. Esto se debe principalmente a que la morfología de la densidad anular ya descrita no tiene un comportamiento exponencial; todos los casos en los que el error no fue mucho mayor que error de ajuste para el polinomio fueron cultivos sin esporulación en los que el comportamiento es más parecido al de la exponencial por no presentar meseta y tener en general un comportamiento monótono creciente.

En el caso de la densidad anular de cultivos con crecimiento homogéneo, los coeficientes del polinomio de ajuste puede proveer de índices que caractericen a cierto tipo de cultivos. Si bien la demostración de lo anterior requiere de estudios adicionales que lo confirmen, en la presente tesis se incluye un ejemplo de la utilización de dichos coeficientes.

La derivada de la densidad anular puede permitir una búsqueda rápida de los puntos de inflexión, para tratar de dividir al cultivo en regiones de acuerdo con los cambios de su densidad anular. En las figuras 22 y 23 se presentan ejemplos de la derivada de la densidad para los cultivos con y sin esporulación de las figuras 20 y 21 respectivamente.



**FIGURA 22.** Gráfica de densidad anular (a) y su derivada (b) para un cultivo típico con esporulación.

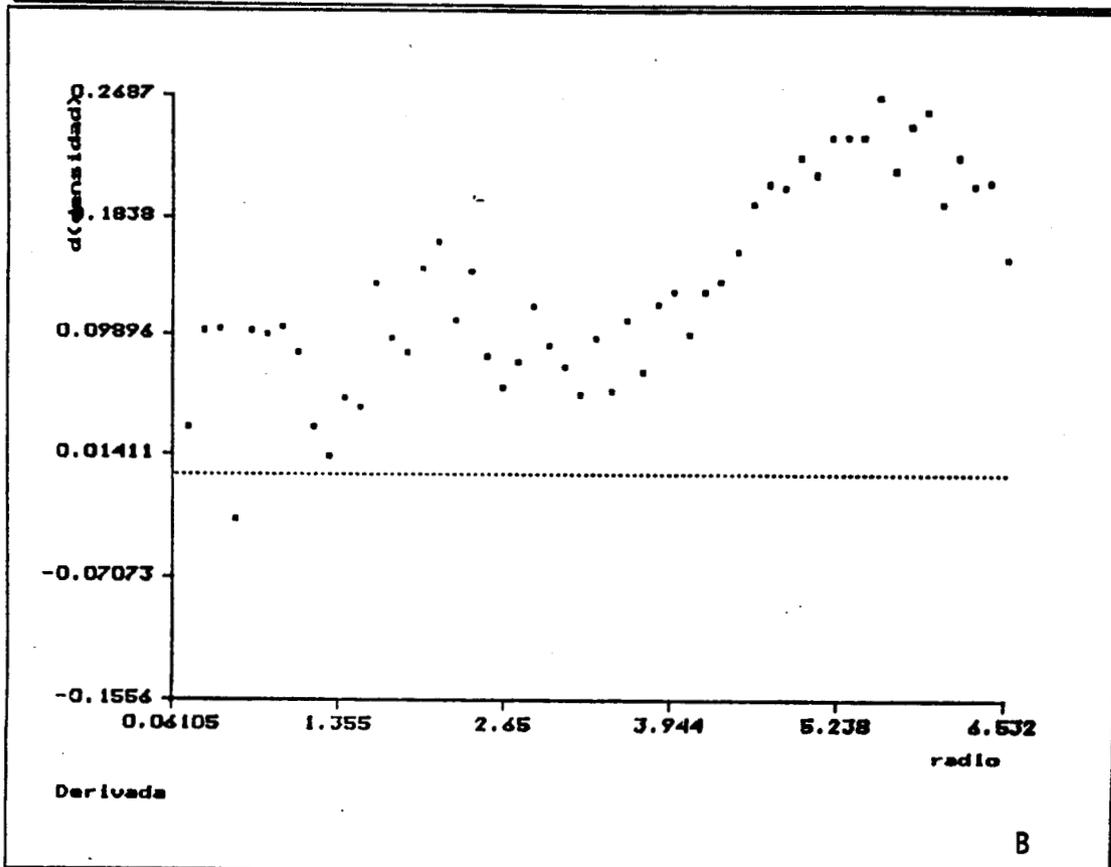
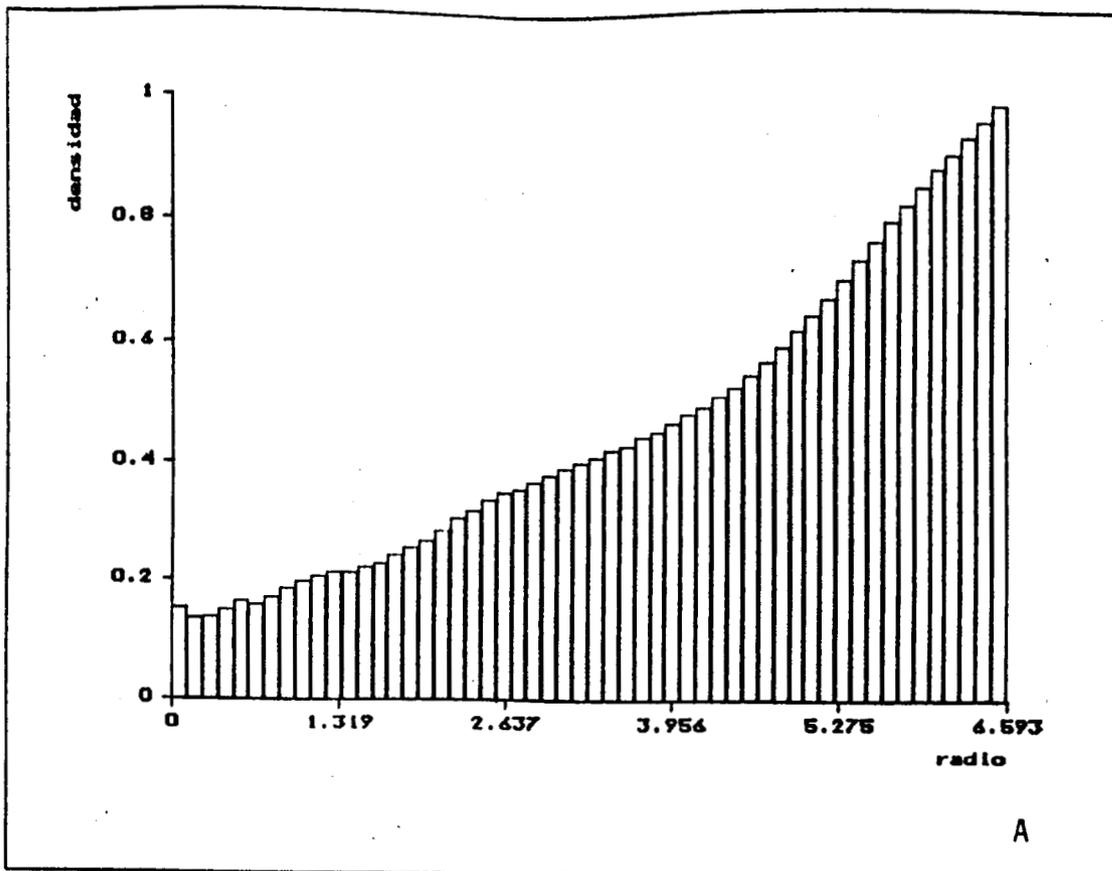


FIGURA 23. Gráfica de densidad anular (a) y su derivada (b) para un cultivo típico sin esporulación.

Si los cultivos que se analizaron hubieran presentado patrones de anillos concéntricos claros y oscuros, la densidad anular reflejaría dichos patrones y mediante la transformada de Fourier hubiera sido posible delimitar características sobre la frecuencia de ocurrencia de dichos anillos; sin embargo, para el conjunto de imágenes analizado la transformada de Fourier no brindó información adicional.

El programa resultó de fácil utilización, principalmente en lo relacionado con la determinación de la densidad anular. Por sus características es posible emplearlo para el análisis de un gran número de cultivos de manera semiautomática (con la supervisión de un operador) y permite el análisis de los resultados de manera interna al programa o utilizando algún otro sistema de procesamiento de información que importe archivos en ASCII.

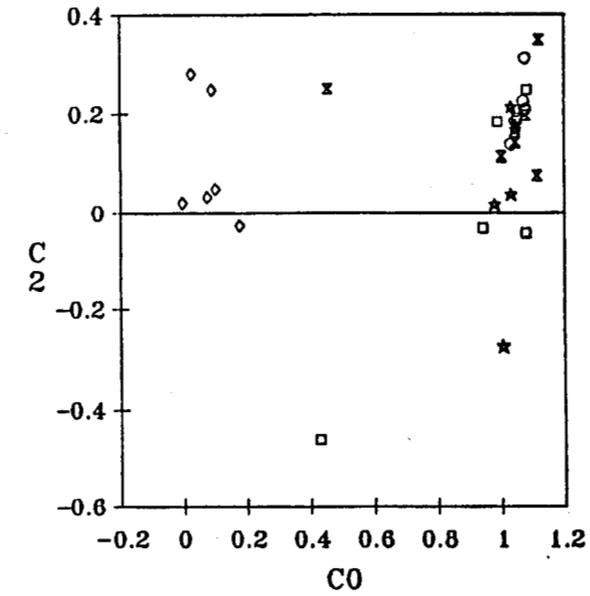
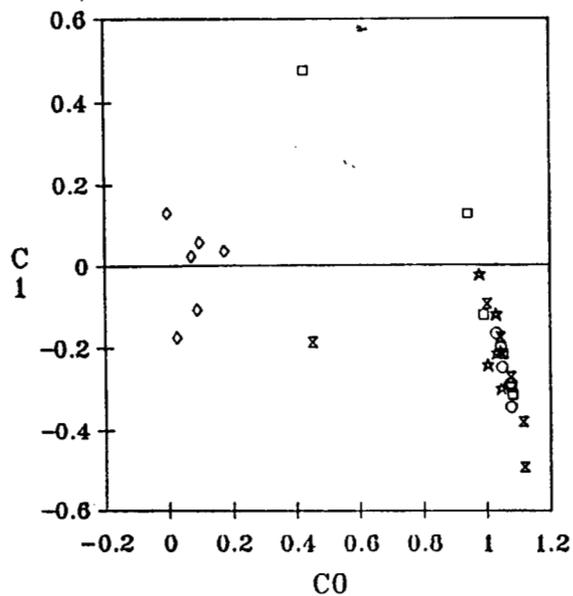
### **Ejemplo de aplicación de la densidad anular.**

La desoxiglucosa es una sustancia tóxica para cierto tipo de mutaciones del hongo *Aspergillus niger*. En el área de microbiología del Departamento de Biotecnología, DCBS. UAM-Iztapalapa se está generando una investigación relacionada con la obtención de mutaciones resistentes a dicho tóxico.

Las cepas utilizadas en los cultivos con los que se probó el sistema forman parte de dicha investigación. La mutación A73 es una cepa patrón, resistente al tóxico, proveniente de Holanda, mientras que las otras cuatro mutaciones fueron generadas en el área de investigación mencionada.

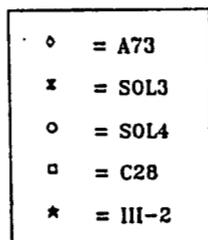
Las imágenes de los treinta cultivos adquiridos de la manera ya descrita corresponden a las cinco mutaciones, inoculadas en medios de cultivo con seis concentraciones diferentes de dicho tóxico (0, 0.01, 0.05, 0.30 y 0.50 gr /L).

Con el fin de ejemplificar el análisis de la densidad del cultivo con los medios propuestos, se obtuvo la densidad anular de los treinta cultivos con los procedimientos ya descritos. El ajuste de la curva de densidad anular se realizó con un polinomio de orden 5 y se utilizaron los primeros tres coeficientes para graficarlos en las proyecciones de un espacio tridimensional (24) y observar el comportamiento de cada mutación para las diferentes concentraciones de desoxiglucosa.

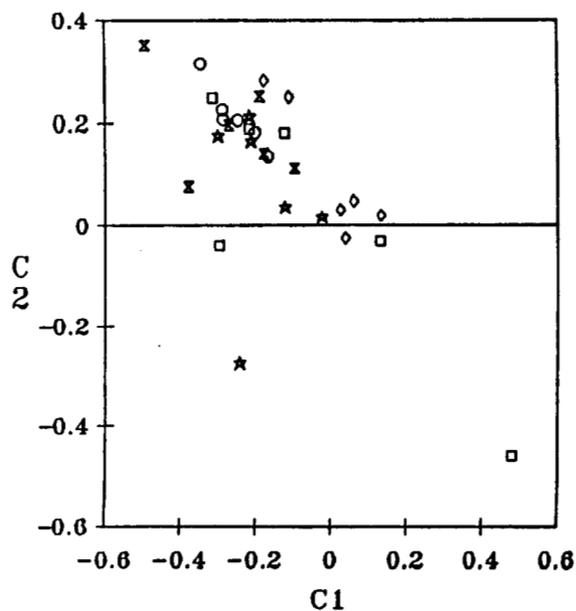


a.

b.



**FIGURA 24.** Gráfica de las proyecciones del espacio tridimensional definido por los primeros tres coeficientes del polinomio de orden 5 ajustado a las curvas de densidad anular de los 30 cultivos de *Aspergillus niger*.



c.

Como se puede observar, existe una separación clara entre los cultivos de las mutaciones generadas en México y el control (A73). En general, se puede observar que aquellos cultivos que produjeron esporulación se diferencian de aquellos que no la presentaron.

Al representar la ordenada al origen, el primer coeficiente del polinomio indica de si existió o no esporulación ya que muestra la densidad relativa para el anillo central (radio interior = 0). En los cultivos con esporulación, este coeficiente es cercano a uno, ya que la zona con más esporulación es la central, mientras que para los cultivos sin esporulación tiene un valor bajo debido al color claro del centro en los cultivos de este tipo.

El cultivo A73 permaneció dentro de una región cercana al origen para las seis concentraciones de desoxiglucosa administradas. Los cultivos de Sol4 y III-2 también presentaron un vector de posición restringido a una región pequeña, pero en este caso con absisas alrededor de uno.

En el caso de los cultivos de Sol3 y C28, para concentraciones bajas de desoxiglucosa, la región fue también restringida y cercana a la región de los cultivos Sol3 y III-2, no así para las altas concentraciones (0.05 gr/L) en las que se pudo observar un corrimiento del primer coeficiente hacia la región central, lo cual indicó una disminución del nivel de esporulación presentado.

Los resultados anteriores pueden indicar que aquellos cultivos que no mostraron corrimientos importantes de su posición en el espacio tridimensional, no presentan variaciones por efecto del tóxico. Sin embargo, estos resultados sólo pueden considerarse como preliminares y no se puede concluir nada definitivo por lo limitado de la muestra y la falta de pruebas de repetibilidad; sólo representan un indicio sobre una de las posibles aplicaciones de este tipo de análisis para la caracterización de cultivos con crecimiento homogéneo.

Todas las imágenes de los cultivos aquí analizados fueron adquiridas para una misma edad del cultivo. Mediante adquisiciones sucesivas a diferentes niveles de desarrollo, es posible que las herramientas aquí planteadas permitan realizar un análisis del crecimiento de este tipo de cultivos con el fin de caracterizarlo para diferentes organismos y mutaciones.

## CONCLUSIONES Y PERSPECTIVAS

El desarrollo de una biblioteca que incluyera procedimientos de uso general para el procesamiento digital de imágenes facilitó el desarrollo de una aplicación específica y puede formar la base para el diseño de nuevas aplicaciones tanto en biotecnología como en otras áreas del conocimiento. Así mismo, puede ser aumentada y actualizada según los requerimientos de dichas aplicaciones.

Los algoritmos diseñados para la determinación y el análisis de la densidad anular resultaron confiables para su aplicación en imágenes de cultivos con crecimiento homogéneo. La generación de un proceso estandarizado para la adquisición de las imágenes permitirá la comparación entre los resultados obtenidos para diferentes cultivos.

Mediante estos procedimientos se pretende realizar la caracterización automática de cientos de mutaciones, utilizando diferentes medios de cultivo, con el fin de obtener los microorganismos óptimos para su utilización en diferentes procesos. Por el volumen de muestras, la cualidad de permitir una detección automática es muy importante. Dado que los dos parámetros determinados automáticamente por el sistema, necesarios para la obtención de la densidad anular, no difieren significativamente de los determinados en forma visual, si se evita lo más posible los problemas planteados para la determinación de éstos dos parámetros se pueden utilizar estos algoritmos para realizar una determinación automática de la densidad.

Por su parte, las técnicas de análisis de la densidad anular resultaron ser un medio adecuado para la caracterización de ciertas cualidades de los cultivos. Estas técnicas pueden utilizarse en un momento dado en forma combinada o generar nuevas técnicas de análisis mediante la creación de índices de las principales características de los cultivos. Además, como la densidad anular puede ser almacenada en disco, en un archivo exportable a una hoja de cálculo comercial, es posible realizar el análisis y crear otras técnicas de análisis en base a los datos proporcionados por los procedimientos diseñados.

Mediante el uso de las herramientas proporcionadas y utilizando diseños experimentales que permitan observar las variaciones a diferentes medios de cultivo, el comportamiento de dichos cultivos a ciertas sustancias

de interés y el análisis temporal del crecimiento, en una comparación transversal y longitudinal, la densidad anular de este tipo de cultivos puede brindar información importante sobre las características morfológicas y temporales de los mismos a nivel macroscópico.

En el Departamento de Biotecnología de la UAM-Iztapalapa se utilizarán los procedimientos planteados para realizar una caracterización de microorganismos con crecimiento homogéneo. Se tratarán de definir características adecuadas, observables por la densidad anular de dichos cultivos, para realizar una clasificación de estos microorganismos.

El campo de la biotecnología es sumamente amplio y la utilización de las imágenes como herramientas en la investigación y en aplicaciones biotecnológicas sigue ampliándose cada vez más. Existen muchos problemas del campo de la biotecnología en los que el procesamiento digital de imágenes puede ser empleado. La utilización de este tipo de procesamiento está extendiéndose tanto en la investigación básica y aplicada como en la monitorización de procesos industriales. Con el desarrollo de sistemas automáticos o semiautomáticos para el procesamiento y análisis de la información obtenible de imágenes, pueden crearse nuevas técnicas de caracterización de organismos y procesos, coadyuvando en la formación de nuevo conocimiento aplicable en la industria de bienes y servicios para el beneficio del ser humano.

## ANEXO 1

### Código de los archivos de definición de los procedimientos incluidos en la biblioteca general de funciones.

En este anexo se presenta el código de definición de los procedimientos incluidos en la biblioteca general de funciones. En él se pueden observar las diversas unidades que conforman la biblioteca, así como los procedimientos, estructuras y macroinstrucciones disponibles para el usuario de la biblioteca.

#### Unidad General

```
/*
    general.h
    Variables globales, estructuras y funciones generales
*/

#if !defined (__GENERAL)
#define __GENERAL

/* interfases requeridas */

#if !defined (__STDIO)
#include <stdio.h>
#endif

#if !defined (__STDLIB)
#include <stdlib.h>
#endif

#if !defined (__MATH)
#include <math.h>
#endif

#if !defined (__DOS)
#include <dos.h>
#endif

#if !defined (__CONIO)
#include <conio.h>
#endif

/* macroinstrucciones */

#define MAXREN      200          /* número máximo de renglones en pantalla */
#define MAXCOL     320          /* número máximo de columnas en pantalla */
```

```

#define TONOS      256          /* número de tonos (niveles de gris) */
#define BLANCO    255          /* nivel de gris del blanco */
#define NEGRO     0            /* nivel de gris del negro */

#define PI        3.141592654  /* número PI */

#define REN      0            /* renglón */
#define COL      1            /* columna */

#define VOID     0            /* tipo de variable void */
#define CH       1            /* tipo de variable char */
#define U_CH     2            /* tipo de variable BYTE */
#define FLO      3            /* tipo de variable float */
#define CMP      4            /* tipo de variable cmplx */
#define INT      5            /* tipo de variable int */

/* estructuras de datos */

typedef struct          /* número complejo */
{
    float re,im;        /* número real e imaginario */
} cmplx;

#if defined(__LARGE__) || defined(__HUGE__) || defined(__COMPACT__)
typedef struct          /* imagen */
{
    void huge *ima;     /* apuntador a los datos */
    unsigned long col;  /* número de columnas */
    unsigned long ren;  /* número de renglones */
    unsigned tipo_var;  /* tipo de datos */
    unsigned tam_var;   /* tamaño del tipo de datos */
} imagen;
#else exit(0);
#endif

typedef unsigned char BYTE;
typedef unsigned      WORD;

typedef struct
{
    float  *x, *y, *y1;
    unsigned numero;} curvas;

/* funciones */

#define round(x) (int)floor(x+0.5)          /* redondea valor de x */
void  espera_tecla (char caracter);        /* espera que se oprima una tecla */
void  ordena_menor (WORD *a, WORD *b);    /* regresa a >, b < */
void  maxmin (imagen imag, float *max, float *min);
                                           /* determina el máximo y el mínimo
                                           de una imagen */
void  max_min_datos(curvas dats, float max_min[2][3]);
                                           /* determina los máximos y mínimos de
                                           dats (x,y,y1) */

```

227465

```

float  obten_valor(void huge *apuntador, WORD tipo);
                                /* obtiene el valor del apuntador */
void   mete_valor(void huge *apuntador, WORD tipo, float valor);
                                /* mete valor en el apuntador */
void huge *apuntador(imagen ima, long x, long y);
                                /* ubica apuntador de ima en x,y */
void   beep(unsigned frec, WORD miliseg); /* da un beep a frec y miliseg */

#endif

```

---

## Unidad Sistema.

```

/*
    sistema.h
    macroinstruccioness y funciones para el manejo del sistema de despliegue
*/

#if !defined (__SISTEMA)
#define __SISTEMA

/* interfases requeridas */

#if !defined (__GENERAL)
#include "general.h"
#endif

#if !defined (__VENT)
#include "ventana.h"
#endif

#if !defined (__STDIO)
#include <stdio.h>
#endif

#if !defined (__STDLIB)
#include <stdlib.h>
#endif

#if !defined (__DOS)
#include <dos.h>
#endif

/* estructuras de datos */

typedef BYTE  PALETA[64][3];

/* macroinstruccioness */

#define MOD_VGA    0x13    /* modo de despliegue gráfico VGA */
#define MOD_TEX   3       /* modo de despliegue en texto */

```

```
/* variables globales */
```

```
extern int mod;  
extern imagen video;
```

```
/* funciones */
```

```
int detecta_modos(void); /* detecta modo de despliegue inicial */  
int cambia_modos(int modo); /* cambia modo de despliegue a modo */  
void paleta(void); /* define la paleta de despliegue */  
void pon_paleta(PALETA pal, BYTE pos); /* define paleta pal y la de alta en el  
segmento pos */
```

```
#endif
```

---

## Unidad Memoria.

```
/*  
    memoria.h  
    Funciones para manejo de memoria de dos  
*/
```

```
#if !defined (__MEMORIA)  
#define __MEMORIA
```

```
/* interfases requeridas */
```

```
#if !defined (__GENERAL)  
#include "general.h"  
#endif
```

```
#if !defined (__STDIO)  
#include <stdio.h>  
#endif
```

```
#if !defined (__STDLIB)  
#include <stdlib.h>  
#endif
```

```
#if !defined (__DOS)  
#include <dos.h>  
#endif
```

```
#if !defined (__ALLOC)  
#include <alloc.h>  
#endif
```

```
/* variables globales */
```

```
extern int mod;  
extern imagen video;
```

**/\* funciones \*/**

```
void mem_video(void);           /* reserva memoria de video */
void borra(imagen *imag);      /* borra imagen de memoria */
imagen define_imagen(unsigned long ren, unsigned long col,
                      unsigned tipo_var, unsigned tam_var);
                              /* define imagen (ren*col) variable
                              tipo_var (reserva memoria) */
```

**#endif**

---

## **Unidad Archivos.**

```
/*
    archivos.h
    Macros y funciones para lectura y escritura de archivos
*/

#if !defined (__ARCHIVOS)
#define __ARCHIVOS

/* interfases requeridas */

#if !defined (__GENERAL)
#include "general.h"
#endif

#if !defined (__SISTEMA)
#include "sistema.h"
#endif

#if !defined (__MEMORIA)
#include "memoria.h"
#endif

#if !defined (__STDIO)
#include <stdio.h>
#endif

#if !defined (__STDLIB)
#include <stdlib.h>
#endif

#if !defined (__CONIO)
#include <conio.h>
#endif

#if !defined (__DIR)
#include <dir.h>
#endif
```

```

#if !defined (__CTYPE)
#include <ctype.h>
#endif

#if !defined (__STRING)
#include <string.h>
#endif

/* macroinstrucciones */

#define EXISTE      1
#define ARCH       2

/* variables globales */

extern imagen video;
extern int mod;

/* funciones */

imagen lectura(void);                /* lee imagen de disco */
void escritura(imagen imag);         /* escribe imagen a disco */
FILE *abre_archivo(char tipo[2], BYTE *format); /* abre archivo del tipo tipo[2] */

#endif

```

---

## Unidad Despliegue.

```

/*
    desplieg.h
    Funciones de despliegue y manejo del cursor
*/

#if !defined (__DESPLIEGUE)
#define __DESPLIEGUE

/* interfases requeridas */

#if !defined (__GENERAL)
#include "general.h"
#endif

#if !defined (__SISTEMA)
#include "sistema.h"
#endif

#if !defined (__MEMORIA)
#include "memoria.h"
#endif

```

```

#if !defined (__STDIO)
#include <stdio.h>
#endif

#if !defined (__STDLIB)
#include <stdlib.h>
#endif

#if !defined (__GRAPHICS)
#include <graphics.h>
#endif

/* variables globales */
extern imagen video;
extern band_color;
extern int mod;

/* estructuras de datos */

typedef struct { unsigned x0;
                unsigned y0;
                unsigned x1;
                unsigned y1;
                } recuadro;

/* macroinstrucciones */

#define DER      1
#define IZQ     2
#define SUB     3
#define BAJ     4

/* tipo de graficas */
#define BARRA    0    /* barras */
#define CIRC1    1    /* y con círculos */
#define RECT2    2    /* y1 con rectángulos */
#define C1R2    3    /* y círculos, y1 rectángulos */
#define CIRC10   4    /* y con círculos, límite inferior = 0 */
#define RECT20   5    /* y1 con rectángulos, límite inferior = 0 */
#define C1R20   6    /* y círculos, y1 rectángulos, límite inferior = 0 */

/* funciones */

void  despliega(unsigned *posv, unsigned *posm, imagen imag, BYTE pal);
      /* despliega imagen CH desde posm en posv */
int   grafica_imagen(unsigned x1, unsigned y1, imagen imag, BYTE pal);
      /* despliega cualquier imagen en x1,y1 */
void  dibuja_cursor(unsigned *posv, BYTE *cursor, BYTE color);
      /* dibuja cursor en posv */
BYTE  mueve_cursor(unsigned *pos_cursor, unsigned *pos_video, unsigned *pos_memo,
                  imagen imag);
      /* mueve el cursor con el teclado */
void  borra_cursor(unsigned *posicion, BYTE *cursor);
      /* borra cursor de la pantalla */

```

```

imagen seccion(int *modo,imagen imag);
                                /* recupera sección (x1-x2,y1-y2) de la
                                imagen de la pantalla */
void  despliega_imagen(unsigned x, unsigned y, imagen imag, BYTE pal);
                                /* despliega imagen CH en posición x,y */
void  graficas(curvas datos, BYTE tipo, char text[4][30]);
                                /* grafica curvas de datos según el tipo */

#endif

```

---

## Unidad Histograma.

```

/*
    histogra.h
    Estructuras y funciones para determinación y manejo de histograma de
    niveles de gris
*/

#if !defined (__HISTO)
#define __HISTO

/* interfases requeridas */

#if !defined (__GENERAL)
#include "general.h"
#endif

#if !defined (__SISTEMA)
#include "sistema.h"
#endif

#if !defined (__DESPLIEGUE)
#include "desplieg.h"
#endif

#if !defined (__STDIO)
#include <stdio.h>
#endif

#if !defined (__STDLIB)
#include <stdlib.h>
#endif

#if !defined (__MATH)
#include <math.h>
#endif

/* macroinstrucciones */

#define DESP 'A'          /* sólo despliega histograma actual */
#define LIN  'B'          /* tmasf. lineal del histograma */
#define NEG  'C'          /* transf. lineal (m<0) del histograma */

```

```

#define LOG 'D' /* transf. logarítmica del histograma */
#define EXP 'E' /* transf. exponencial del histograma */
#define SEN 'F' /* transf. senoidal (0-PI) del histograma */
#define SEN2 'G' /* transf. senoidal (0-PI/2) del histograma */
#define COS 'H' /* transf. cosenoidal (0-PI) del histograma */
#define COS2 'I' /* transf. cosenoidal (0-2PI) del histograma */
#define EQU 'J' /* igualación del histograma */
#define ESP 'K' /* especificación del histograma */

#define TAM_HISTO 40 /* tamaño de la gráfica de histograma */
#define TAM_BARRA 5 /* tamaño de la barra de gama */

/* variables globales */

extern imagen video; /* imagen de la pantalla */
extern int mod; /* modo de despliegue anterior*/
extern int band_color; /* pseudocolor o grises */
extern PALETA *altas_paleta[5]; /* paletas definidas */

/* estructuras de datos */

typedef struct
{
    unsigned long *histo; /* histograma */
    int max,min; /* valores máximo y mínimo */
}arr_histo; /* arreglo histograma */

/* funciones */

void histograma_g(imagen imag, arr_histo *niveles);
/* determina el histograma de niveles de gris */
void dibuja_histo_g(arr_histo niveles,unsigned x,unsigned y,BYTE color);
/* dibuja histograma de niveles de gris */
void transf_histo_g(imagen imag, char tipo, unsigned con_paleta);
/* transformación del histograma por mapa */

/* transformaciones */
void transf_lineal(arr_histo transf, unsigned mx, unsigned mn);
/* elongación lineal */
void transf_log(arr_histo transf, unsigned mx, unsigned mn);
/* logarítmica */
void transf_exp(arr_histo transf, unsigned mx, unsigned mn);
/* exponencial */
void transf_senPI(arr_histo transf, unsigned mx, unsigned mn);
/* senoidal de 0 a PI */
void transf_senPI2(arr_histo transf, unsigned mx, unsigned mn);
/* senoidal de 0 a PI/2 */
void transf_cosPI(arr_histo transf, unsigned mx, unsigned mn);
/* cosenoidal de 0 a PI */
void transf_cos2PI(arr_histo transf, unsigned mx, unsigned mn);
/* cosenoidal de 0 a 2PI */
void transf_equ(arr_histo transf, arr_histo orig, unsigned tam_ima);
/* igualación del histograma */

```

```
void  transf_esp(arr_histo transf, unsigned mx, unsigned mn);
/* especificación del histograma */
void  menu_histo(imagen imag);

#endif
```

---

## Unidad Segmenta.

```
/*
    segmenta.h
    Funciones de segmentación de imágenes y semitonos
*/

#if !defined (__SEGMENTA)
#define __SEGMENTA

/* interfases requeridas */

#if !defined (__GENERAL)
#include "general.h"
#endif

#if !defined (__DESPLIEGA)
#include "despliega.h"
#endif

#if !defined (__SISTEMA)
#include "sistema.h"
#endif

#if !defined (__STDIO)
#include <stdio.h>
#endif

#if !defined (__CONIO)
#include <conio.h>
#endif

#if !defined (__STDLIB)
#include <stdlib.h>
#endif

/* macroinstrucciones */

#define S_BLANCO 0
#define S_NEGRO 1
#define S_GRIIS 2
#define S_GRIIS_N 3
```

```
/* variables globales */
```

```
extern PALETA *altas_paleta[5];  
extern imagen video;  
extern int modo;  
extern int band_color;
```

```
/* funciones */
```

```
void segmenta (int nivel[2],imagen imag, BYTE tipo);  
/* segmenta la imagen por nivel */  
void segmenta_gama (imagen imag,BYTE tipo);  
/* referencia de gama de la imagen */  
BYTE segmenta_p(BYTE pal,int *umbral,BYTE tipo);  
/* segmenta por manejo de paleta */  
void menu_segmenta(imagen ima); /* menu para la segmentación */
```

```
#endif
```

---

## Unidad Bordes.

```
/*
```

```
    bordes.h
```

```
    Funciones para el realce de bordes
```

```
*/
```

```
#if !defined (__BORDES)  
#define __BORDES
```

```
/* interfases requeridas */
```

```
#if !defined (__GENERAL)  
#include "general.h"  
#endif
```

```
#if !defined (__STDIO)  
#include <stdio.h>  
#endif
```

```
#if !defined (__STDLIB)  
#include <stdlib.h>  
#endif
```

```
#if !defined (__MATH)  
#include <math.h>  
#endif
```

```
/* variables globales */
```

```
extern int mod;  
extern imagen video;
```

**/\* funciones \*/**

```
void  gradiente(imagen ima);           /* borde por gradiente */
void  aprox_max(imagen ima, unsigned delta); /* borde por aproximación al gradiente
                                           por maximo */
void  aprox_sum(imagen ima, unsigned delta); /* borde por aproximación al gradiente
                                           por suma */
void  laplaciano(imagen ima);          /* borde por laplaciano */
void  detecta_bordes(imagen ima);     /* menu de selección */
void  realce(imagen ima);              /* imagen menos laplaciano */
```

**#endif**

---

## **Unidad Filtros.**

```
/*
    filtros.h
    Estructuras y funciones para utilizar los filtros
*/
```

```
#if !defined (__FILTRO)
#define __FILTRO
```

**/\* interfases requeridas \*/**

```
#if !defined (__GENERAL)
#include "general.h"
#endif
```

```
#if !defined (__SISTEMA)
#include "sistema.h"
#endif
```

```
#if !defined (__STDIO)
#include <stdio.h>
#endif
```

```
#if !defined (__STDLIB)
#include <stdlib.h>
#endif
```

```
#if !defined (__MEMORIA)
#include "memoria.h"
#endif
```

```
#if !defined (__CONIO)
#include <conio.h>
#endif
```

```
/* variables globales */
```

```
extern int mod;  
extern imagen video;
```

```
/* funciones */
```

```
imagen define_ventana(void); /* define ventana para filtro promedio */  
void filtro_mediana(unsigned tam_vent, imagen imag);  
/* filtro de mediana con ventana cuadrada */  
void filtro_promedio(imagen pesos, imagen imag);  
/* filtro promediador con pesos en filtro */  
void filtro_sigma(unsigned tam_vent, unsigned nsigma, imagen imag);  
/* filtro sigma que abarca nsigma de */  
void filtro_kvecinos(unsigned tam_vent, unsigned k, imagen imag);  
/* filtro de k próximos vecinos */  
void filtro_grad_inv(unsigned n, imagen imag);  
/* filtro gradiente inverso, ventana 2n+1 */  
void filtrado(imagen imag); /* menú de selección */
```

```
#endif
```

---

## Unidad MediMorf.

```
/*  
    medimorf.h  
    Funciones para realizar mediciones morfológicas  
*/
```

```
#if !defined (__MEDIMORF)  
#define __MEDIMORF
```

```
/* interfases requeridas */
```

```
#if !defined (__GENERAL)  
#include "general.h"  
#endif
```

```
#if !defined (__SISTEMA)  
#include "sistema.h"  
#endif
```

```
#if !defined (__ARCHIVOS)  
#include "archivos.h"  
#endif
```

```
#if !defined (__STDIO)  
#include <stdio.h>  
#endif
```

```
#if !defined (__STDLIB)
#include <stdlib.h>
#endif
```

```
/* variables globales */
```

```
extern imagen video;
extern int mod;
extern int band_color;
```

```
/* funciones */
```

```
float largo(int *modo, imagen imag);           /* largo entre cursores */
float area_st_circulo(int *modo, imagen imag); /* area de sección transversal */
float area_elipse(int *modo, imagen imag);     /* area de elipse */
float volumen_cilindro(int *modo, imagen imag); /* volumen con forma cilíndrica */
float mediciones(int *modo, imagen imag);      /* mediciones en la imagen */
float calibracion(void);                       /* determina factor de calibración */
```

```
#endif
```

---

## ANEXO 2

### Programación desarrollada para la obtención y el análisis de la densidad anular de cultivos con crecimiento homogéneo.

Se presenta la programación original desarrollada para la aplicación del procesamiento digital de imágenes en la determinación de la densidad anular de cultivos con crecimiento homogéneo.

#### Unidad Anillos.

```
/*
    anillos.h
    Estructuras y funciones para determinar la densidad anular de un cultivo.
*/

#if !defined (__ANILLOS)
#define __ANILLOS

/* interfases requeridas */

#if !defined (__GENERAL)
#include "general.h"
#endif

#if !defined (__SISTEMA)
#include "sistema.h"
#endif

#if !defined (__DESPLIEGUE)
#include "desplieg.h"
#endif

#if !defined (__METNUM)
#include "metnum.h"
#endif

#if !defined (__FFT_1D)
#include "fft_1d.h"
#endif

#if !defined (__HISTOGRAM)
#include "histograma.h"
#endif

#if !defined (__MEDIMORF)
#include "medimorf.h"
#endif

#if !defined (__ARCHIVOS)
#include "archivos.h"
#endif
#endif
```

```
#if !defined (__STDIO)
#include <stdio.h>
#endif
```

```
#if !defined (__STDLIB)
#include <stdlib.h>
#endif
```

```
#if !defined (__MATH)
#include <math.h>
#endif
```

```
#if !defined (__CONIO)
#include <conio.h>
#endif
```

```
#if !defined (__ALLOC)
#include <alloc.h>
#endif
```

```
#if !defined (__CTYPE)
#include <ctype.h>
#endif
```

```
#if !defined (__STRING)
#include <string.h>
#endif
```

```
/* macroinstrucciones */
```

```
#define DERB 1          /* define zona de graficado */
#define DERS 2
#define IZQB 3
#define IZQS 4
```

```
/* estructuras de datos*/
```

```
typedef struct          /* definición de un anillo */
{
    float r1;          /* radio interior */
    float r2;          /* radio exterior */
    float area;        /* área del anillo en pixeles */
    float nivel;       /* nivel promedio del anillo */
    float dens;        /* densidad del anillo */
} densidad;
```

```
typedef struct
{
    densidad *dens;    /* apuntador al arreglo dens */
    unsigned num_anillos; /* numero de anillos */
} arr_dens;
```

**/\* variables globales \*/**

```
extern int mod;
extern imagen video;
extern int band_color;
extern float F_CAL;
```

**/\* funciones \*/**

```
void    centroide(imagen imag, recuadro cuadro, unsigned *centro);
        /* centro de gravedad del cultivo */
void    centro_med_min(imagen imag, recuadro cuadro, unsigned *centro);
        /* centro de ventana 3x3 con med min */
unsigned centro_x_minimo(imagen imag, recuadro cuadro, unsigned *centro);
        /* centro = dx mínima */
void    centro_min_dx_dy(imagen imag, recuadro cuadro, unsigned *centro);
        /* centro por mínimo del Laplaciano */
void    densidad_anular(imagen imag);          /* determina densidad anular y la grafica */
unsigned calcula_densidad_anular(arr_dens *dens,imagen imag, BYTE tipo);
        /* determina la densidad anular */
void    grafica_dens(arr_dens dens);          /* grafica la densidad anular */
```

#endif

---

## Programación Anillos.

**/\* ANILLOS.C \*/**

#include "anillos.h"

```
float max_densidad(arr_dens dens);
unsigned detecta_tam_cult(unsigned centro[2], recuadro cuad);
void tdf_dens (arr_dens dens);
void dibuja_diametro(unsigned *posv,unsigned *tam_cult);
void dibuja_centro(unsigned *pos_c,imagen imag);
void detecta_centro(imagen imag, unsigned *pos_c, BYTE tipo, recuadro *cuad);
void normaliza_dens (arr_dens dens);
void grafica_dens(arr_dens dens);
BYTE menu_centro(void);
BYTE menu_trat_anular(void);
void transforma_anillos(arr_dens dens);
curvas mete_datos_dens(arr_dens dens, BYTE tipo);
void centro_recuadro(recuadro cuadro, unsigned *centro);
void dibuja_cuadro(recuadro cuad);
void salvar_datos(arr_dens dens);
```

```
/* _____ Rutina principal de determinación de la densidad anular _____ */
```

```
void densidad_anular(imagen imag)
{
    arr_dens     dens;
    unsigned     tam_cult;
    int          modo;
    BYTE         op;
    char         textos[4][30];
    VENTANA TRABAJANDO = { 38, 12,
                          30, 3,
                          LIGHTGRAY, WHITE, ""};
    VENTANA DENSIDAD = { 40, 10,
                        23, 11,
                        LIGHTMAGENTA, WHITE,
                        " DENSIDAD ANULAR "};

    if((!imag.ima)&&(imag.tipo_var == U_CH))
    {
        if((modo=detecta_modo())!=MOD_TEX) cambia_modo(MOD_TEX);
        do
        {
            abre_ventana(&DENSIDAD);
            op = menu_centro();           /* define tipo de algoritmo de detección del centro */
            if(op!=27)
            {
                abre_ventana(&TRABAJANDO);
                cprintf(" Elongando Histograma ");           /* elongación lineal del hitograma de grises */
                transf_histo_g(imag,LIN,0);
                cierra_ventana(&TRABAJANDO);
                tam_cult = calcula_densidad_anular(&dens,imag,op); /* determina la densidad anular */
                dens.num_anillos = tam_cult;
                normaliza_dens(dens);           /* normalización */
                grafica_dens(dens);
                transforma_anillos(dens);      /* procedimientos de análisis */
                salvar_datos(dens);             /* almacenar archivo de densidad en disco */
                free((void *)dens.dens);
            }
            cierra_ventana(&DENSIDAD);
        }
        while(op!=27);
        cambia_modo(modo);
    }
}
```

```

BYTE menu_centro(void)                                /* opciones de detección del centro */
{
    BYTE op;

    cprintf("\n\n");
    cprintf(" Detector de centro:\n\n");
    cprintf(" 1. Centroide \n\n");
    cprintf(" 2. dx m;nimo \n\n");
    cprintf(" 3. Media m;nima\n\n");
    cprintf(" 4. dx,dy m;nimo \n\n");
    cprintf(" 5. Recuadro \n\n");
    cprintf(" <ESC> Salir \n\n");
    do
        op = getch();
    while((op!='1')&&(op!='2')&&(op!='3')&&(op!='4')&&(op!='5')&&(op!=27));
    return(op);
}

/* _____ Determina la densidad anular _____ */

unsigned calcula_densidad_anular(arr_dens *dens, imagen imag, BYTE tipo)
{
    unsigned    x,y,pos_c[2];          /* pos_c[0]=x, pos_c[1]=y */
    unsigned    i,j,k,xo,yo,tam_cult=0;
    float       r;
    recuadro    cuad;
    VENTANA TRABAJANDO = { 38, 12,
                          30, 3,
                          LIGHTGRAY, WHITE, ""};

    if((imag.ima)&&(imag.tipo_var == U_CH))
    {
        detecta_centro(imag,pos_c,tipo,&cuad);          /* detecta centro */
        x = pos_c[0];
        y = pos_c[1];
        if(abs(x) < abs(x-imag.col)) xo = imag.col - x - 1;
        else xo = x;
        if(abs(y) < abs(y-imag.ren)) yo = imag.ren - y - 1;
        else yo = y;
        tam_cult = detecta_tam_cult(pos_c,cuad);        /* determina tamaño del cultivo */
        cambia_modos(MOD_TEX);
        abre_ventana(&TRABAJANDO);
        cprintf(" Calculando densidad ");
        dens->num_anillos = floor(sqrt(xo*xo + yo*yo));    /* inicialización de variables */
        dens->dens =(densidad *)malloc(dens->num_anillos *sizeof(densidad));
    }
}

```

```

if(dens->dens)
{
for(i=0;i<dens->num_anillos;i++)
{
(dens->dens)[i].r1 = i*F_CAL;
(dens->dens)[i].r2 = (i+1)*F_CAL;
(dens->dens)[i].area = 0;
(dens->dens)[i].nivel = 0;
(dens->dens)[i].dens = 0;
}
for(j=0;j<imag.ren;j++) /* cálculo de las variables necesarias*/
for(i=0;i<imag.col;i++) /* para determinar la densidad anulr */
{
r = sqrt((i-x)*(i-x) + (j-y)*(j-y));
k = round(r);
(dens->dens)[k].area += F_CAL*F_CAL;
(dens->dens)[k].nivel += (BLANCO -
((BYTE huge *)imag.ima)[j*imag.col+i]);
}
for(i=0;i<dens->num_anillos;i++) /* cálculo de densidad no normalizada */
{
if((dens->dens)[i].area)
(dens->dens)[i].dens = (dens->dens)[i].nivel/(dens->dens)[i].area;
else dens->num_anillos = i;
}
cierra_ventana(&TRABAJANDO);
}
}
return(tam_cult);
}

```

/\* ----- detección del centro del cultivo ----- \*/

```

void detecta_centro(imagen imag, unsigned *pos_c, BYTE tipo, recuadro *cuad)
{
BYTE cursor[9];
BYTE c_blanco;
unsigned c_recuad[2];
VENTANA TRABAJANDO = { 38, 12,
30, 3,
LIGHTGRAY, WHITE, ""};

cuad->x0 = cuad->y0 = 0;
cuad->x1 = imag.col-1;
cuad->y1 = imag.ren-1;
zona_centro(imag,cuad); /* determina región de búsqueda del centro */
abre_ventana(&TRABAJANDO);
cprintf(" Detectando centro ");
}

```

```

switch (tipo)
{
    case '1': centroide(imag,*cuad,pos_c);break;
    case '2': centro_x_minimo(imag,*cuad,pos_c);break;
    case '3': centro_med_min(imag,*cuad,pos_c);break;
    case '4': centro_min_dx_dy(imag,*cuad,pos_c);break;
    case '5': centro_recuadro(*cuad,pos_c);break;
}
cierra_ventana(&TRABAJANDO);
cambia_modos(MOD_VGA);
despliega_imagen(0,0,imag,1+band_color);
dibuja_cuadro(*cuad);
dibuja_centro(pos_c,imag);
dibuja_cursor(pos_c,cursor,1);
}

```

**/\* Algoritmos de detección : \*/**

```

void centroide(imagen imag, recuadro cuadro, unsigned *centro)
{
    long tonos_act, sx=0, sy=0, sn=0;
    unsigned i,j;
    if(imag.ima)
    {
        for(i=cuadro.x0;i<=cuadro.x1;i++)
            for(j=cuadro.y0;j<=cuadro.y1;j++)
            {
                tonos_act = BLANCO - ((BYTE huge *)imag.ima)[j*imag.col + i];
                sx += i*tonos_act;
                sy += j*tonos_act;
                sn += tonos_act;
            }
        centro[0] = sx/sn;
        centro[1] = sy/sn;
    }
}

```

```

void centro_med_min(imagen imag, recuadro cuadro, unsigned *centro)
{
    unsigned i,j,k,l,min = BLANCO;
    int med_ant;
    BYTE c_blanco;
#define delta 1
    if(imag.ima)
    {
        centro_recuadro(cuadro,centro);
        c_blanco =
            (((BYTE huge *)imag.ima)[centro[1]*imag.col+centro[0]]>TONOS/2)?1:0;
        centro[0]=centro[1]=cuadro.x0;
    }
}

```

```

for(j=cuadro.y0+delta;j<cuadro.y1-delta;j++) /* recorre la zona de búsqueda*/
for(i=cuadro.x0+delta;i<cuadro.x1-delta;i++)
{
med_ant = 0;
for(k=j-delta;k<=j+delta;k++) /* calcula media de la ventana */
for(l=i-delta;l<=i+delta;l++)
med_ant += ((BYTE huge *)imag.ima)[k*imag.col + l];
if(c_blanco) med_ant = BLANCO*(delta*2+1)*(delta*2+1) - med_ant;
if(min >= med_ant) /* comparación */
{
min = med_ant;
centro[0] = i;
centro[1] = j;
}
}
}
}

unsigned centro_x_minimo(imagen imag, recuadro cuadro, unsigned *centro)
/* por min o max en la dirección X */
{
unsigned i,j,min = BLANCO;
int dif_ant,dif_sig;
BYTE c_blanco;

if(imag.ima)
{
centro_recuadro(cuadro,centro); /* determina si hay esporulación */
c_blanco =
(((BYTE huge *)imag.ima)[centro[1]*imag.col+centro[0]]>TONOS/2)?1:0;
centro[0]=centro[1]=cuadro.x0;
for(j=cuadro.y0;j<cuadro.y1;j++) /* recorre zona de búsqueda */
{
dif_ant = ((BYTE huge *)imag.ima)[j*imag.col+cuadro.x0+1] -
((BYTE huge *)imag.ima)[j*imag.col+cuadro.x0];
for(i=cuadro.x0+1;i<cuadro.x1;i++)
{
dif_sig = ((BYTE huge *)imag.ima)[j*imag.col+i+1] -
((BYTE huge *)imag.ima)[j*imag.col+i];
if(dif_ant*dif_sig < 0) /* verifica cambio de pendiente */
if(min >= BLANCO*c_blanco +
pow(-1,c_blanco)*((BYTE huge *)imag.ima)[j*imag.col+i])
/* comparación */
{
min = BLANCO*c_blanco +
pow(-1,c_blanco)*((BYTE huge *)imag.ima)[j*imag.col+i];
centro[0] = i;
centro[1] = j;
}
}
dif_ant = dif_sig;
}
}
}
return(min);
}

```

```

void centro_min_dx_dy(imagen imag, recuadro cuadro, unsigned *centro)
{
    unsigned i,j, *c;
    int dy1,dy2,minx,miny=BLANCO;
    BYTE c_blanco;

    if(imag.ima)
    {
        centro_recuadro(cuadro,centro); /* detrmina si hay esporulación */
        c_blanco =
            (((BYTE huge *)imag.ima)[centro[1]*imag.col+centro[0]]>TONOS/2)?1:0;
        minx = centro_x_minimo(imag, cuadro, centro); /* determinación en dirección X */
        c[0]=cuadro.x0;
        c[1]=cuadro.y0; /* determinación en dirección Y */
        for(i=cuadro.x0;i<=cuadro.x1;i++) /* recorre la zona de búsqueda */
        {
            dy1 = ((BYTE huge *)imag.ima)[i] -
                ((BYTE huge *)imag.ima)[imag.col+i];
            for(j=cuadro.y0;j<cuadro.y1;j++)
            {
                dy2 = ((BYTE huge *)imag.ima)[(j+1)*imag.col+i] -
                    ((BYTE huge *)imag.ima)[j*imag.col+i];
                if(dy1*dy2 < 0)
                if(miny >= BLANCO*c_blanco +
                    pow(-1,c_blanco)*((BYTE huge *)imag.ima)[j*imag.col+i])
                {
                    miny = BLANCO*c_blanco +
                        pow(-1,c_blanco)*((BYTE huge *)imag.ima)[j*imag.col+i];
                    c[0] = i;
                    c[1] = j;
                }
                dy1 = dy2;
            }
        }
        if(minx>miny) /* comparación entre direcciones */
        {
            centro[0] = c[0];
            centro[1] = c[1];
        }
    }
}

void centro_recuadro(recuadro cuadro, unsigned *centro)
{
    /* por centro de la zona de búsqueda */
    centro[0] = (cuadro.x0 + cuadro.x1)/2;
    centro[1] = (cuadro.y0 + cuadro.y1)/2;
}

```

```

/*----- supervisión por operador -----*/
void dibuja_centro(unsigned *pos_c,imagen imag)
{
  char op;
  unsigned pos_v[3],pos_m[2],x,y;

  x = pos_c[0];          /* inicializa posición del cursor en centro */
  y = pos_c[1];
  pos_v[0]=pos_v[1]=0;
  pos_v[1]=1;
  pos_m[0]=pos_m[1]=0;
  op = mueve_cursor(pos_c,pos_v,pos_m,imag);      /* permite mover el centro detectado */
  if(op!=13)
  {
    pos_c[0] = x;
    pos_c[1] = y;
  }
}
/*-----*/

```

```

/*-----Delimitación de la zona de búsqueda -----*/

void zona_centro(imagen ima, recuadro *cuad)
{
  imagen aux;          /* limita la zona de búsqueda del centro */
  BYTE band_disco = 0;
  BYTE band_sube = 0;
  BYTE band_izq = 0;
  unsigned z[2];
  BYTE n=5;
  recuadro cuad1;
  VENTANA TRABAJANDO = { 38, 12,
                        30, 3,
                        LIGHTGRAY, WHITE, ""};

  aux = define_imagen(ima.ren,ima.col,ima.tipo_var,ima.tam_var);
  /* protege información original */
  if(aux.ima)
    memmove((void *)aux.ima,(void *)ima.ima,ima.ren*ima.col);
  else
  {
    band_disco = a_disco(ima);
    aux.ima = ima.ima;
    aux.ren = ima.ren;
    aux.col = ima.col;
    aux.tipo_var = ima.tipo_var;
    aux.tam_var = ima.tam_var;
  }
}

```

```

if((aux.ima)||band_disco)
{
  abre_ventana(&TRABAJANDO);
  cprintf(" Resaltando bordes ");
  transf_histo_g(aux,SEN2,0);
  gotoxy(1,1);
  cprintf("Delimitando zona de centro ");
  do
  {
    z[0]=z[1]=0;
    busca_borde(aux,z,DERB,&n);
    cuad->x0 = z[0];
    cuad->y0 = z[1];
    z[1]=aux.ren-1;
    z[0]=aux.col-1-band_izq*(aux.col -1);
    if(!band_izq)busca_borde(aux,z,IZQS,&n);
    cuad->x1 = z[0];
    cuad->y1 = z[1];
    if((n>1)&&(!band_sube))n--;
    else {band_sube = 1; n++;}
  }
  while((cuad->x1<cuad->x0)||((cuad->y1<cuad->y0));
  band_sube = 0;
  n = 5;
  do
  {
    z[0]=aux.col-1;
    z[1]=0;
    busca_borde(aux,z,IZQB,&n);
    band_izq = 1;
    cuad1.x1 = z[0];
    cuad1.y0 = z[1];
    z[1]=aux.ren-1;
    z[0]=aux.col-1-band_izq*(aux.col -1);
    busca_borde(aux,z,DEFS,&n);
    cuad1.x0 = z[0];
    cuad1.y1 = z[1];
    if((n>1)&&(!band_sube))n--;
    else {band_sube = 1; n++;}
  }
  while((cuad1.x1<cuad1.x0)||((cuad1.y1<cuad1.y0));
  if(cuad->x0 < cuad1.x0) cuad->x0 = cuad1.x0;
  if(cuad->x1 > cuad1.x1) cuad->x1 = cuad1.x1;
  if(cuad->y0 < cuad1.y0) cuad->y0 = cuad1.y0;
  if(cuad->y0 > cuad1.y1) cuad->y0 = cuad1.y1;
  if(!band_disco)
  original */
  borra(&aux);
  else
  de_disco(ima);
  cierra_ventana(&TRABAJANDO);
}
}

```

*/\* resalta borde del cultivo \*/  
 /\* por manejo del histograma de grises\*/*

*/\* busca borde en de las trayectorias \*/*

*/\* mientras no sea adecuado \*/*

*/\* busca borde en otra trayectoria \*/*

*/\* mientras no sea adecuado \*/  
 /\* calcula menor rectángulo \*/*

*/\* recupera imagen*

void busca\_borde(imagen ima, unsigned \*z, BYTE tipo, BYTE \*n)

```

{
/* busca borde en una trayectoria a 45° por
umbral variable */
unsigned i=0,max;
float valor,med=0,de=0,Sx2;
BYTE borde=0;

if(ima.ima)
{
max = (ima.ren>ima.col)?ima.col : ima.ren;
while((i<max-1)&&!borde) /* mientras no borde */
{
valor = (float)((BYTE huge *)ima.ima)[z[1]*ima.col + z[0]];
if((valor <= abs(med)+(*n)*de)||i<5) /* compara con umbral variable */
switch(tipo) /* tipo de trayectoria */
{
case DERB : {z[0]++;z[1]++;}break;
case DERS : {z[0]++;z[1]--;}break;
case lZQB : {z[0]--;z[1]++;}break;
case lZQS : {z[0]--;z[1]--;}break;
}
else borde = 1;
i++; /* actualiza parámetros del umbral */
med = (med*(i-1) + valor)/i;
Sx2 = (Sx2*(i-1) + valor*valor)/i;
de = sqrt(Sx2 - med*med);
}
}
}

BYTE a_disco(imagen imag) /* guarda imagen temporalmente */
{
unsigned i;
BYTE nombre[15]="tempima.ima";
FILE *archivo = NULL;
void huge *temp;

archivo = fopen(nombre,"w+"); /* abre el archivo */
if(archivo)
{
temp = imag.ima;
for(i=0;i<imag.ren;i++) /* recorre imagen */
{
fwrite((void *)temp,imag.tam_var,imag.col,archivo); /* escribe */
(BYTE huge *)temp += imag.col*imag.tam_var;
}
fclose(archivo); /* cierra archivo */
return(1);
}
return(0);
}

```

```

void de_disco(imagen imag)          /* Lee Imagen temporal de Disco */
{
FILE          *archivo;           /* archivo de lectura */
BYTE          nombre[15]="tempima.ima";
unsigned long i;
void huge     *temp;

archivo = fopen(nombre,"r+");      /* abre el archivo */
if(archivo)
{
/* define imagen de salida */
if(imag.ima)                       /* si hubo espacio */
{
temp = imag.ima;
for(i=0;i<imag.ren;i++)           /* recorre archivo */
{
fread((void *)temp,imag.tam_var,imag.col,archivo); /* lee imagen */
((BYTE huge *)temp) += imag.col*imag.tam_var;
}
}
else mensaje(" No recuperó la imagen ");
fclose(archivo);                  /* cierra archivo */
}
else mensaje(" No recuperó la imagen ");;
}

void dibuja_cuadro(recuadro cuad)
{
/* dibuja región de búsqueda sobre la imagen */
unsigned i;

for(i=cuad.x0;i<=cuad.x1;i++)
{
((BYTE huge *)video.ima)[cuad.y0*video.col + i] = 63*3;
((BYTE huge *)video.ima)[cuad.y1*video.col + i] = 63*3;
}
for(i=cuad.y0;i<=cuad.y1;i++)
{
((BYTE huge *)video.ima)[i*video.col + cuad.x0] = 63*3;
((BYTE huge *)video.ima)[i*video.col + cuad.x1] = 63*3;
}
}

```

```
/* ----- Determina el tamaño del cultivo ----- */
```

```
unsigned detecta_tam_cult(unsigned centro[2], recuadro cuad)
{
    /* determina el tamaño del cultivo (radio) */
    unsigned tamano;
    float radio2;

    radio2 = ((cuad.x0-centro[0])*(cuad.x0-centro[0])+
              (cuad.x1-centro[0])*(cuad.x1-centro[0])+
              (cuad.y0-centro[1])*(cuad.y0-centro[1])+
              (cuad.y1-centro[1])*(cuad.y1-centro[1]))/2;    /* promedio de distancia entre centro
                                                             y vértices de zona de búsqueda */

    tamano = round(sqrt(radio2));    /* cálculo del radio en pixeles */
    dibuja_diametro(centro,&tamano);    /* supervisión del operador */
    return(tamano);
}
```

```
void dibuja_diametro(unsigned *posv,unsigned *tam_cult)
{
    unsigned i,tam_ini,tam_max;
    unsigned minx=0,miny=0;
    char tecla=0,ext;
    BYTE *lineax,*lineay;
#define AZUL 1
#define ROJO 62

    if(detecta_modos()==MOD_VGA)
    {
        /* dibuja radios iniciales en ambas direcciones */
        tam_ini = *tam_cult;
        tam_max = sqrt(posv[0]*posv[0] + posv[1]*posv[1]);
        lineax = (BYTE *)malloc(MAXCOL*sizeof(BYTE));
        lineay = (BYTE *)malloc(MAXREN*sizeof(BYTE));
        if((lineax)&&(lineay))
        {
            for(i=0;i<MAXCOL;i++)
                lineax[i] = ((BYTE huge *)video.ima)[posv[1]*MAXCOL + i];
            for(i=0;i<MAXREN;i++)
                lineay[i] = ((BYTE huge *)video.ima)[i*MAXCOL + posv[0]];
            while((tecla!=13)&&(tecla!=27))
            {
                /* mientras el operador no seleccione un radio final */
                memmove((void *)((BYTE huge *)video.ima+posv[1]*MAXCOL), (void *)lineax,MAXCOL);
                for(i=0;i<MAXREN;i++)
                    ((BYTE huge *)video.ima)[i*MAXCOL+posv[0]] = lineay[i];
                if(posv[0] > *tam_cult) minx = posv[0]-(*tam_cult);
                if(posv[1] > *tam_cult) miny = posv[1]-(*tam_cult);
                for(i=minx;i<posv[0]+(*tam_cult);i++)
                    ((BYTE huge *)video.ima)[posv[1]*MAXCOL+i] = ROJO+64*(2-band_color);
                for(i=miny;i<posv[1]+(*tam_cult);i++)
                    ((BYTE huge *)video.ima)[i*MAXCOL+posv[0]] = ROJO+64*(2-band_color);
            }
        }
    }
}
```

```

tecla = getch();
if(!tecla) ext = getch();
switch (ext)
{
case 72 : if(*tam_cult<tam_max)(*tam_cult)++;break;
case 80 : if(*tam_cult>0)(*tam_cult)--;break;
case 0 : if((tecla != 27)&&(tecla != 13)) beep(2000,50);break;
default : beep(2000,50);break;
}
}
if(tecla==27)*tam_cult = tam_ini; /* tecla = <ESC> radio automático */
memmove((void *)((BYTE huge *)video.ima+posv[1]*MAXCOL),
(void *)lineax,MAXCOL);
for(i=0;i<MAXREN;i++)
((BYTE huge *)video.ima)[i*MAXCOL+posv[0]] = lineay[i];
for(i=minx;i<posv[0]+(*tam_cult);i++)
((BYTE huge *)video.ima)[posv[1]*MAXCOL+i] = AZUL+64*(2-band_color);
for(i=miny;i<posv[1]+(*tam_cult);i++)
((BYTE huge *)video.ima)[i*MAXCOL+posv[0]] = AZUL+64*(2-band_color);
free((void *)lineax);
free((void *)lineay);
}
}
}

```

**/\* ----- Almacenamiento y despliegue de la densidad ----- \*/**

```

void salvar_datos(arr_dens dens)
{
char *text_out=NULL; /* almacena densidad anular en disco */
char text_num[50];
BYTE op,i,f[4];
FILE *archivo;
VENTANA48Y VENTANA SALVAR = { 40, 12,
39, 3,
BLUE, YELLOW, ""};
VENTANA TEXTO = { 30, 14,
50, 3,
LIGHTGRAY, BLUE, "ENCABEZADO"};

if(dens.dens)
{
abre_ventana(&SALVAR);
printf(" Guardar la densidad anular (S/N) : ");
cursor_on();
do
op = toupper(getch());
while((op!='S')&&(op!='N'));
}
}

```



```

free(dat.x);          /* libera memoria */
free(dat.y);
free(dat.y1);
}
}

/*----- Procesamiento y análisis de la densidad -----*/

BYTE menu_trat_anular(void)          /* menu de opciones */
{
  BYTE op;

  printf("\n\n");
  printf(" (P)olinomial\n\n");
  printf(" (E)xponencial\n\n");
  printf(" (G)ausiano\n\n");
  printf(" (D)erivada\n\n");
  printf(" (F)ourier\n\n");
  printf("<ESC> Ninguno\n\n");
  do
    op = toupper(getch());
  while((op!='P')&&(op!='E')&&(op!='G')&&(op!='D')&&(op!='F')&&(op!=27));
  return(op);
}

void transforma_anillos(arr_dens dens) /* realiza algún procesamiento sobre la
                                       densidad */
{
  curvas      dat;
  unsigned    i,orden;
  BYTE        op;
  char        textos[4][30];
  VENTANA TRATAMIENTO = { 44, 14,
                          20, 10,
                          BLUE, YELLOW,
                          " TRATAMIENTO "};
  VENTANA ORDEN = { 44, 14,
                   28, 3,
                   LIGHTGRAY, BLUE, ""};

  if(dens.dens)
  {
    abre_ventana(&TRATAMIENTO);
    do
    {
      gotoxy(1,1);
      op = menu_trat_anular();          /* pregunta tipo de tratamiento */
      if(op!=27)
      {
        dat = mete_datos_dens(dens,C1R2); /* datos a estructura "curva" */
      }
    }
  }
}

```

```

switch(op)
{
  case 'P' : {
      abre_ventana(&ORDEN);
      cursor_on();
      cprintf(" Orden del polinomio : ");
      scanf("%d",&orden);
      cursor_off();
      cierra_ventana(&ORDEN);
      ajuste_polinomio(orden,NORM,dat);
    }break;
  case 'E' : ajuste_exp(dat);break;
  case 'G' : ajuste_gaussiano(dat);break;
  case 'D' : {
      derivada(dat,1);
      dat.numero--;
      strcpy(textos[0],"radio");
      strcpy(textos[1],"d(densidad)");
      strcpy(textos[2],"Derivada");
      strcpy(textos[3],"");
      graficas(dat,RECT2,textos);
    }break;
  case 'F' : tdf_dens(dens);break;
}
free(dat.x);
free(dat.y);
free(dat.y1);
}
}while (op!=27);
cierra_ventana(&TRATAMIENTO);
}
}

void tdf_dens (arr_dens dens)
{
  unsigned i;
  senal s;

  s.sn1 = (cplx far *)farmalloc(dens.num_anillos*sizeof(cplx));
  if(s.sn1)
  {
    s.dim = dens.num_anillos;
    for(i=0;i<s.dim;i++)
    {
      s.sn1[i].re = dens.dens[i].dens;
      s.sn1[i].im = 0;
    }
    FFT(DIR,&s);
    magnitud_cuad(s);
    grafica_frec(s);
  }
  else
  mensaje("No se puede determinar la TDF");
}

```

```
/* _____ funciones generales _____ */
```

```
void normaliza_dens(arr_dens dens)
```

```
{  
    /* normaliza densidad con el valor máximo */  
    unsigned i;  
    float max;  
  
    if(dens.dens)  
    {  
        max = max_densidad(dens);  
        for(i=0;i<dens.num_anillos;i++)  
            dens.dens[i].dens = dens.dens[i].dens/max;  
    }  
}
```

```
curvas mete_datos_dens(arr_dens dens, BYTE tipo)
```

```
{  
    /* pasa a una estructura "curvas" la información  
    de la densidad anular */  
  
    unsigned i;  
    curvas dat;  
  
    dat.x = dat.y = dat.y1 = NULL;  
    if(dens.dens)  
    {  
        dat.numero = dens.num_anillos;  
        dat.x = (float *)malloc(sizeof(float)*dat.numero);  
        dat.y = (float *)malloc(sizeof(float)*dat.numero);  
        dat.y1 = (float *)malloc(sizeof(float)*dat.numero);  
        if((dat.x)&&(dat.y)&&(dat.y1))  
        {  
            if(tipo==BARRA)  
                for(i=0;i<dat.numero;i++)  
                {  
                    dat.x[i] = (float)dens.dens[i].r1;  
                    dat.y[i] = dat.y1[i] = dens.dens[i].dens;  
                }  
            else  
                for(i=0;i<dat.numero;i++)  
                {  
                    dat.x[i] = (float)(dens.dens[i].r2+dens.dens[i].r1)/2;  
                    dat.y[i] = dat.y1[i] = dens.dens[i].dens;  
                }  
        }  
    }  
    return(dat);  
}
```

```
float max_densidad(arr_dens dens)
{
    unsigned i;
    float valor,maximo=0;

    if(dens.dens)
        for(i=0;i<dens.num_anillos;i++)
        {
            valor = dens.dens[i].dens;
            if(maximo<valor) maximo = valor;
        }
    return(maximo);
}
```

/\* determina el máximo del arreglo densidad \*/

## **BIBLIOGRAFIA**

1. Adams L, Constant D. Biosterometrics in the study of the morphology of the lumbar sacral spine. *Med Biol Eng Comp* 1988; 26:383-8.
2. Adams H, Thomas C. The use of image analysis for morphological measurements on filamentous microorganisms. *Biotechnol Bioeng* 1988; 32:707-12.
3. Albe X, Cornu MB, Margules S, Bisconte JC. Software package for the quantitative analysis of histological sections. *Computers and Biomedical Research* 1985; 18:313-33.
4. Bjorsen PK. Automatic determination of bacterio-plankton biomass by image analysis. *Appl Environ Microbiol* 198; 51:1199-204.
5. Blackstad TW, Karagülle T, Ottersen OP. MORFOMEL, a computer program for two-dimensional analysis of micrographs of biological specimens, with emphasis on immynogold preparations. *Comput Biol Med* 1990; 20:15-34.
6. Bradbury S. Microscopical image analysis: problems and approaches. *J Microsc* 1979; 115:137-50.
7. Brigham RN. The fast Fourier transform and its applications. Englewood Cliffs N.J.: Prentice Hall, 1977.
8. Camp CE, Sublette KL. Control of a Thiobacillus denitrificants bioreactor using machine vision. *Biotechnol Bioeng* 1992; 39:529-38.
9. Chapra SC, Canale RP. Métodos numéricos para ingenieros con aplicaciones en computadoras personales. EUA: McGraw-Hill, 1985.

10. Costello PJ, Monk PR. Image analysis method for the rapid counting of *Sccharomyces cervisiae* cells. *Appl Environ Microbiol* 1985; 49:863-6.
11. D'Amarato RF, et al. Novel application of video image processing to biochemical and antimicrobial susceptibility testing. *J Clin Microbiol* 1988; 26:1492-5.
12. Ford RM, Phillips BR, Quin JA, Lauffenburger DA. Measurement of bacterial random motility and chemotaxis coefficients. I. Stopped-floe diffusion chamber assay. *Biotechnol Bioeng* 1991; 37:647-60.
13. Fry JC, Davies AR. An assessment of methods for measuring volumes of placktonic bacteria, with particular reference to TV image analysis. *J Appl Bacteriol* 1985; 58:105-12.
14. Garza-Jinich M, Rodriguez-Carranza C, Corkidi G, Toledo R, Montero R, Ostrosky P. Supervised mitotic index scoring as a tool in cytotoxicity studies. *Memorias de Vision Interface 91: Canadian Image Processing and Patter Recognition Society*; 1991.
15. González RC, Woods RE. *Digital image processing*. EUA: Adison Wesley, 1992.
16. Gunde-Cimerman N, Cimerman A, Perdith A. *Aspergillus niger* mutants for bioconversion of apple distillery wastes. *Enzyme Microb Technol* 1986; 8:166-169.
17. He, L. Chai Z. New method of atomatic SCE detection. *Pattern Recognit* 1990; 23:95-7.
18. Hochmuth G. *Métodos cinematográficos de investigación en Biomecánica de los movimientos depotivos*. Madrid, DONCEL, 1973. pp 161-178.

19. Hoover LK, Moo-Young M, Legge RL. Biotransformation of dopamine to norlaudanosoline by *Aspergillus niger*. *Biotechnol Bioeng* 1991; 38:1029-33.
20. Huls PG, Nanninga N, van Spronsen EA, Valkenburg JAC, Vischer NOE, Woldring CL. Computer-aided measuring system for the characterization of yeast populations combining 2D-image analysis, electronic particle counter, and flow cytometry. *Biotechnol Bioeng* 1992; 39:343-50.
21. Kaczmarek E. Tree partition algorithm for analysis of morphological structure of material. *Appl Math Modeling* 1989; 13:584-9.
22. Katsinis C, Poularikas AD. Analysis of a sampling technique applied to biological images. *IEEE Trans Patt Analysis and Machine Intell* 1987; 9:832-5.
23. Kéry V, Kogan G, Zajacova K, Slamova K, Masler L, Alfoeldi J. Hydrolysis of yeast cell-wall glucan by extracellular (1->3)-BETA-glucanases from *Aspergillus niger*. *J Enzyme Microb Technol* 1991; 13:87-90.
24. Koehler JK. (Editor). *Advanced techniques in biology electron microscopy*. Nueva York: Springer-Verlag, 1973.
25. Leuchtenberg A, Mayer G. Changed pectinase synthesis by aggregated mycelium of some *Aspergillus niger* mutants. *Enzyme Microb Technol* 1992; 14:18-22.
26. Lira J. *Caracterización matemática de imágenes digitales*. Comunicaciones Técnicas Instituto de Geofísica, UNAM.
27. Lord PG, Wheals AE. Asymmetrical division of *S. cerevisiae*. *J Bacteriol* 1980; 142:808-18.
28. Luhtanen P, Komi PV. Segmental contribution to forces in vertical jump. *Eur J Appl Physiol* 1978; 38:181-8.

29. Mason DT, Demaria AN, Berman DS. Técnicas no invasivas en cardiología. Ecocardiografía y cardiología nuclear. Barcelona: Ediciones Doyma, 1982.
30. Merrell BR, Weiss E, Dasch GA. Morphological and cell association characteristics of *Rochalimaea quintana*: Comparison of the vole and fuller strains. *J Bacteriology*, 1978. 135:633-40.
31. Metz B, De Bruijn E, van Suijdam J. Method for quantitative representation of the morphology of molds. *Biotechnol Bioeng* 1981; 23:149-62.
32. Mostafavi, H. An imagen sequence analysis workstation for multi-point motion analysis. Abstracts of the First World Congress of Biomechanics, 1990: 9.
33. Murr LE, Berry VK. Direct observations of selective attachment of bacteria on low-grade sulfide ores and other mineral surfaces. *Hydrometallurgy* 1976; 2:11-24.
34. Packer HL, Keshavorz-Moore E, Lilly MD, Thomas CR. Estimation of cell volume and biomass of *penicillium chrysogenum* using image analysis. *Biotechnol Bioeng* 1992; 39:384-91.
35. Packer H, Thomas C. Morphological measurements on filamentous microorganisms by fully automatic image analysis. *Biotechnol Bioeng* 1990; 35:870-81.
- mechanics 1977; 10:133-140.
36. Peli T, Malah D. A study of edge detection algorithms. *Comput Graphics Image Processing* 1982; 20:1-21.
37. Pettipher GL, Rodrigues UM. Semi-automated counting of bacteria and somatic cells in milk using epifluorescence microscopy and television image analysis. *J Appl Bacteriol* 1982; 53:323-9.

38. Popechitelev EP, Soloviev NI. Designing test patterns for checking TV analyzers. *Meas Tech* 1990; 33:629-31.
39. Reed IS, Glenn WV, Truong TK, Kwoh YS, Chang CM. X-Ray reconstruction of the spinal cord, using bone suppression. *IEEE Trans on Biomedical Engineering* 1980; BME27:293-8.
40. Reichl U, Buschulte T, Gilles E. Study of the early growth and branching of *Streptomyces tendae* by means of an image processing system. *J Microsc* 1990; 158:55-62.
41. Reichl U, King R, Guilles ED. Characterization of pellet morphology during submerged growth of *Streptomyces tendae* by image analysis. *Biotechnol Bioeng* 1992; 39:164-70.
42. Reichl U, Yang H, Gilles ED, Wolf H. An improved method for measuring the interseptal spacing in hyphae of *Streptomyces tendae* by fluorescence microscopy coupled with image processing. *FEMS Microbiol Lett* 1990; 67:207-10.
43. Roehr M, Zehentgruber O, Kubicek CP. Kinetics of biomass formation and citric acid production by *Aspergillus niger* on pilot plant scale. *Biotech Bioeng* 1981; 23:2433-55.
44. Saltin B, Henriksson J, Nygaard E, Andersen P. Fiber types and metabolic potentials of skeletal muscles in sedentary man and endurance runners. *N.Y. Acad Sci* 1977; 301:3-29.
45. Sjollem J, et al. Real-time enumeration of adhering microorganisms in a parallel plate flow cell using automated image analysis. *J Microbiol Meth* 1989; 9:73-8.
46. Skaer AVG, Skaer RJ. A guidebook of microscopical methods. London Cambridge at the University Press, 1972.

47. Tahoces PG, Correa J, Souto M, González C, Gómez L, Vidal JJ. Enhancement of chest and breast radiographs by automatic spatial filtering. *IEEE Trans Med Imaging* 1991; 10:330-5.
48. Takahashi J. Sistema semiautomático para detectar y evaluar lesiones de esclerosis múltiple mediante imágenes de resonancia nuclear magnética. *Rev Mex Ing Biomed* 1989; 10:1-10
49. Toledo R, Usisima R, Marquez J, Corkidi G. Valoración automatizada de las células del endotelio corneal humano por procesamiento digital de imágenes. *Rev Mex Ing Biomed* 1990; 11:81-7.
50. Villanueva M. Manual de técnicas somatotipológicas. México: Universidad Autónoma de México, Serie antropológica: 31; 1979.
51. Yanagita T, Kogané F. Growth and cytochemical differentiation of mold colonies. *J Gen Appl Microbiol* 1962; 8:201-13.
52. Yanagita T, Kogané F. Cellular differentiation of growing mold colonies with special reference to phosphorus metabolism. *J Gen Appl Microbiol* 1963; 9:313-30.
53. Yang H, King R, Reichl U, Guilles ED. Mathematical model for apical growth, septation and branching of mycelial microorganism. *Biotechnol Bioeng* 1992; 39:49-58.
54. Yang H, Reichl U, King R, Guilles ED. Measurement and simulation of the morphological development of filamentous microorganism. *Biotechnol Bioeng* 1992; 39:44-8.