

03063

6
Lej



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

U. A. C. P. y P.
I. I. M. A. S.

PROPUESTA DE UN MÉTODO DE
SEGMENTACIÓN DE IMÁGENES
BASADO EN LAS TEXTURAS

TESIS

Que para obtener el Grado de

MAESTRA EN CIENCIAS DE LA COMPUTACIÓN

presenta

MARÍA GUADALUPE LÓPEZ MOLINA

Director de tesis:

M.C. SALVADOR BERNARDO MARTÍNEZ JIMÉNEZ

México, D.F.

Octubre de [REDACTED]

TESIS CON
FALLA DE ORDEN

1999



273168



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

INDICE

i

INTRODUCCIÓN

iii

CAPITULO I: PROCESAMIENTO DIGITAL DE IMÁGENES

1

1.1 Areas de interés

4

1.2 Historia

5

1.3 Etapas fundamentales del procesamiento de imágenes.

7

1.4 Sistemas de procesamiento digital de imágenes.

11

1.5 Conceptos básicos.

16

1.6 Segmentación.

20

1.7 Técnicas de segmentación.

23

**CAPITULO II. MARCO TEORICO: SEGMENTACION BASADA EN
TEXTURAS.**

27

2.1 Texturas

27

2.2 Co-ocurrencia de niveles de gris

32

2.3 Forma a partir de la textura

35

2.4 Modelos de mosaicos aleatorios

36

2.5 Bordes de las texturas

37

2.6 Transformaciones ortogonales

37

2.7 Densidad de los extremos relativos

40

2.8 Autorregresión

43

2.9 Morfología matemática

45

2.10 Campos aleatorios markovianos discretos

46

2.11 Función de autocorrelación

48

2.12 Dispersión de vectores

49

2.13 Energía de las texturas	50
2.14 Segmentación basada en texturas	51
CAPITULO III: DESCRIPCION DE UN METODO PARA SEGMENTACION BASADA EN TEXTURAS.	54
3.1 Energia de las texturas	55
3.2 Uso de suavizado de características y relajación probabilística	58
3.3 Método de vertientes	64
3.4 Métodos de relajación para segmentación supervisada de Imágenes	72
CAPITULO IV: IMPLANTACION Y RESULTADOS	75
4.1 Primera: Realce	75
4.2 Segunda: Suavizado	76
4.3 Tercera: Media controlada	77
4.4 Cuarta: Agrupamiento y etiquetado probabilístico inicial	79
4.5 Quinta: Relajación	82
CONCLUSIONES	93
BIBLIOGRAFÍA.	95
ANEXO 1: TABLAS DE DESTINO UTILIZADAS	102
ANEXO 2: CODIGO FUENTE	107

INTRODUCCION

En diferentes áreas de trabajo es necesario dividir una imagen sobre la base de la textura que ésta tiene. Un par de ejemplos de las aplicaciones en que se usa la segmentación basada en la textura es distinguir en una fotografía tomada por medio de satélite zonas erosionadas de aquellas que no lo están, o bien, en el ultrasonido de una mujer embarazada distinguir el feto y la placenta de la piel y el intestino.

A pesar de que diversos grupos de investigación en todo el mundo han trabajado por muchos años en segmentación basada en las texturas el problema no se ha resuelto completamente por lo que esta línea de estudio sigue vigente.

El objetivo de este trabajo es proponer un nuevo método para la segmentación basada en la textura que en primer lugar, mejore los resultados de la segmentación de imágenes con textura, especialmente a lo largo de las fronteras y en segundo lugar, mejore la segmentación de los interiores de las regiones, tomando en cuenta las relaciones espaciales de los pixeles.

Respecto a los alcances y limitaciones de este trabajo es conveniente mencionar que se pretende presentar un método, para lo que es necesario mostrar una implantación del mismo, sin embargo, el producto principal de este trabajo no es un sistema.

A lo largo de cinco años se ha ido reuniendo material respecto al tema, pero fue el hecho de que se publicara el artículo de Hansen y Higgins en 1997 lo que reavivó el interés de hacer de esta idea un trabajo terminal de la Maestría en Ciencias de la Computación e incorporar alguna de sus propuestas en este trabajo. A pesar de que no se sigue a la letra el algoritmo propuesto por ellos, sus ideas ayudaron a cortar un "nudo gordiano", que se había encontrado hace algunos años.

Ya que se ha mencionado la motivación de este estudio, se considera conveniente describir brevemente el contenido de los capítulos que componen este trabajo en los párrafos siguientes:

En el capítulo uno se presenta lo referente al procesamiento digital de imágenes y a la segmentación de imágenes en lo general. En lo que respecta al procesamiento digital de imágenes se narra brevemente su historia y se mencionan algunas de las áreas en la que el tratamiento de información "visual" es muy necesario. Asimismo, en este capítulo se explican algunos de los conceptos básicos que serán utilizados a lo largo de este trabajo.

El segundo capítulo contiene la información referente a las texturas de una imagen y las principales familias de métodos de análisis de texturas que existen. Esto con la finalidad de proporcionar un marco teórico a este trabajo. Se explica de manera breve pero completa la esencia de cada una de estas familias de métodos y se da cuenta de algunos de los trabajos publicados en cada una de ellas. Se habla de familias de métodos y no sólo de métodos debido a que en cada una de estas categorías se incluyen procedimientos similares pero con características distintivas.

El capítulo tres describe el método propuesto en este trabajo. Para ello fue necesario discutir más a detalle algunos de los métodos de análisis de texturas y de segmentación basada en texturas que han sido elegidos para formar parte del método propuesto.

La implantación de éste método se describe en el cuarto capítulo. En este se encuentra la descripción detallada de cada una de las etapas que componen al método. También se encuentran ahí los resultados de aplicar este método y la comparación con los resultados obtenidos por el método de Hsiao y Sawchuk (1989a y 1989b).

Finalmente se encuentran las conclusiones de este trabajo y algunas recomendaciones respecto a líneas a seguir y a trabajo futuro.

CAPITULO I: PROCESAMIENTO DIGITAL DE IMÁGENES

Una línea actual dentro de la investigación en computación, es la relacionada con el procesamiento digital de imágenes. Este término se refiere generalmente a la serie de técnicas que se aplican a una imagen bidimensional haciendo uso de una computadora. El procesamiento digital de imágenes tiene una amplia variedad de aplicaciones, entre las que lo más obvio es mencionar la percepción remota por medio de satélites, la transmisión y almacenamiento de imágenes para aplicaciones administrativas, la astronomía, el procesamiento de imágenes médicas, imágenes de radar, de sonar, visión por computadora usada en robótica e inspección de partes industriales.

El procesamiento digital de imágenes puede ser usado para detectar recursos naturales, trazar mapas geográficos más precisos, hacer predicciones sobre las cosechas, el crecimiento urbano o el clima, y muchas otras aplicaciones ambientales. En las imágenes obtenidas de misiones espaciales, por medio de procesamiento digital de imágenes, se mejora su calidad y se realiza el reconocimiento y análisis de objetos. Las aplicaciones de transmisión y almacenamiento de imágenes incluyen la automatización de oficinas, comunicación en redes de computadoras, y sistemas de monitoreo basados en televisión de circuito cerrado, entre otras. Las aplicaciones médicas abarcan el procesamiento de radiografías, angiogramas, tomografías y muchas otras aplicaciones en radiología, citología y ultrasonido.

Además, el procesamiento digital de imágenes puede ser usado para el monitoreo de pacientes o para detección de tumores o enfermedades. El radar y el sonar usan el procesamiento digital de imágenes para la detección de varios tipos de blancos o para guiar y maniobrar aeronaves o sistemas de misiles. Hay muchas otras aplicaciones que varían desde la visión de robot para automatización industrial hasta la síntesis de imagen para la realización de caricaturas o diseño de modas.

Dentro de la variedad de aplicaciones aquí mencionadas, pueden distinguirse dos áreas que si bien son diferentes, están muy relacionadas por los autores que trabajan en el campo. El procesamiento de imágenes, según Sid-Ahmed (1995), está relacionado con el mejoramiento de imágenes para la percepción humana. La visión artificial, por otro lado se refiere al procesamiento de imágenes para el análisis de la máquina. Un ejemplo del primero es corregir una imagen desenfocada que una persona verá. Para aclarar el segundo se puede pensar el reconocimiento de una escena que necesita hacer un robot para poder moverse.

Castleman (1996) define el procesamiento digital de imágenes como el someter la representación numérica de un objeto a una serie de operaciones para obtener el resultado deseado. En el caso de fotografías, el proceso cambia su forma para hacerlas más deseables o atractivas o para lograr alguna otra meta predefinida. De acuerdo con este autor, el procesamiento digital de imágenes principia con una imagen y produce una versión modificada de ella. Es por tanto, un proceso que lleva de una imagen a otra imagen. El análisis digital de imágenes, por otra parte, significa un proceso que lleva una imagen digital a algo diferente que puede ser un conjunto de medidas de tamaño o forma o incluso una decisión. Por ejemplo, si una imagen digital contiene una serie de objetos, un programa puede analizar la imagen y extraer las medidas como la forma y el tamaño de dichos objetos.

A pesar de que se suelen diferenciar el procesamiento y el análisis de imágenes, también es común que el término procesamiento digital de imágenes sea usado de manera laxa para englobar a ambos.

Ya que el tema tratado es el procesamiento digital de imágenes, es conveniente definir una imagen en términos de este estudio. Una imagen digital se

define como un arreglo bidimensional rectangular de valores de muestra, a cada uno de los elementos del arreglo se le llama pixel.

Haralick y Shapiro (1992), en su libro *Computer and Robot Vision* definen una imagen como la representación espacial de un objeto, una escena bi o tri dimensional u otra imagen. Esta puede ser real o virtual como en óptica. En la visión por computadora la imagen usualmente significa imagen grabada, tal como una imagen de vídeo, una imagen digital o una fotografía.

Más formalmente, el término imagen monocroma o simplemente imagen se refiere a una función bidimensional de intensidad de luz $f(x, y)$, donde x e y representan las coordenadas espaciales y el valor de f en un punto cualquiera (x, y) es proporcional al brillo (o nivel de gris) de la imagen en ese punto. A veces es útil imaginar una imagen como una función en perspectiva en la cual el tercer eje representa el brillo. Si se asignan valores proporcionalmente más altos a las áreas más brillantes, se lograría que la altura de las diversas partes de esta representación sea proporcional al correspondiente brillo de la imagen.

Una imagen digital es una imagen $f(x, y)$ a la que se le han asignado valores discretos tanto a las coordenadas espaciales como al brillo. Una imagen digital puede considerarse como una matriz cuyos índices de renglón y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Los elementos de una distribución digital de este tipo se denominan elementos de la imagen, o más comúnmente pixeles (abreviaturas de su denominación inglesa "*picture elements*").

Aunque el tamaño de una imagen digital varía dependiendo de su aplicación, se suelen seleccionar matrices cuadradas con tamaños y número de niveles de gris que sean potencias enteras de dos. Esto se debe principalmente a la forma de funcionamiento binaria de las computadoras. Por ejemplo, un tamaño

típico, comparable en calidad a una imagen monocroma de televisión, es una matriz de 512 x 512 puntos con 128 niveles de gris.

1.1 AREAS DE INTERES.

Las aplicaciones mencionadas con anterioridad pueden ser clasificadas de diferentes formas. A pesar de que diferentes autores han clasificado las áreas de interés del procesamiento digital de imágenes en diferentes categorías se suelen reconocer al menos las cuatro categorías siguientes: realce, compresión, restauración y reconocimiento de patrones.

a) Realce de imágenes. En muchos casos deseamos procesar las imágenes para mejorar su calidad. Un ejemplo de esta categoría son las fotografías provenientes de las sondas espaciales. Estas imágenes, tal como llegan, son difíciles de entender, debido a que durante su envío han sufrido distorsiones. El realce permite quitar o cuando menos suavizar esta distorsión, de tal forma que al comparar las imágenes de antes y después del procesamiento de realce, las diferencias suelen ser sorprendentes. Este procesamiento busca mejorar sustancialmente la calidad de varios tipos de información de la imagen (por ejemplo los bordes), generalmente en un contexto de percepción.

b) Compresión de imágenes. Cuando una imagen requiere para su transmisión y almacenamiento de una capacidad excesiva, es interesante codificarla, de alguna manera, para reducir este requerimiento. Para ejemplificar esto, usaremos la necesidad de guardar imágenes de rayos X en los hospitales. Como el número de las imágenes que es necesario conservar es enorme, su almacenamiento requiere de mucho espacio. Esto nos lleva a la necesidad de una aplicación que reduzca el espacio que requiere cada imagen para ser almacenada, pero que conserve las características que la hacen valiosa.

c) Restauración de imágenes. Se refiere a la corrección de imágenes borrosas o con otros efectos de distorsión introducidos durante el proceso de formación de la imagen. Usualmente se hace respecto a una medida objetiva de fidelidad. Dentro de este campo se incluye la restauración de imágenes desenfocadas por tres tipos de causas: imágenes con mal enfoque al ser guardadas, imágenes borrosas por estar "movidas" e imágenes borrosas debido a que han estado expuestas mucho tiempo a la intemperie.

d) Reconocimiento de patrones. En esta área se pretende decidir sobre la presencia o ausencia de un objeto o clase de objetos en una escena de la imagen y estimar ciertos parámetros de los objetos. Usamos el término reconocimiento de patrones para referirnos a la detección y extracción de objetos, texturas u otra información de las imágenes. Un ejemplo de los problemas que se atacan actualmente en este campo, es el reconocimiento de caracteres escritos a mano, otro es el reconocimiento de patrones que ayude al diagnóstico médico, un tercer ejemplo es el trazo automático de mapas de recursos naturales utilizando los datos de fotografías de satélites y discriminando las áreas que componen a la fotografía.

1.2 HISTORIA.

El interés en el procesamiento digital de imágenes se remonta a principios de la década de los años veinte, cuando las fotografías digitalizadas de las noticias mundiales se transmitían por cable submarino. Sin embargo, no fue sino hasta los años sesenta cuando, con los avances logrados en electrónica, las computadoras empezaron a ofrecer las capacidades de almacenamiento y rapidez requeridas para la implementación práctica de algoritmos de procesamiento de imágenes.

En 1964 se usaron algunas técnicas computacionales para mejorar las fotografías provenientes de la luna que eran transmitidas por el satélite Ranger 7,

para mejorar distorsiones inherentes a la cámara que este llevaba. Desde entonces, esta área ha experimentado un crecimiento vigoroso y ha sido objeto de estudio e investigación interdisciplinarios en varios campos que aparentemente no están relacionados, pero que comparten la necesidad de métodos capaces de mejorar información pictórica para su interpretación y análisis.

Desde la década pasada, el impacto de las técnicas computacionales en todos los campos de las ciencias de la información ha sido profundo y difundido. Esto es, desde luego, cierto en el campo del procesamiento digital de imágenes, donde han sido empleadas casi exclusivamente las aproximaciones digitales a la manipulación de imágenes y otros conjuntos de datos bidimensionales en un amplio rango de aplicaciones científicas.

El procesamiento digital de imágenes ha resultado ser una parte integral de aplicaciones tales como tomografía computarizada, exploración geofísica, pruebas de materiales no destructivas, astronomía, ultrasonido médico, percepción remota y robótica industrial. De hecho es difícil conceptualizar muchas de estas aplicaciones sin sus componentes de procesamiento de imágenes.

El desarrollo de algoritmos más sofisticados de procesamiento digital de imágenes ha sido un elemento importante dentro de la evolución del campo. A este respecto ha existido una relación de "estira y afloja" entre los componentes de hardware y el desarrollo de algoritmos. Conforme se han expandido las capacidades del procesador, se han desarrollado nuevos algoritmos para explorar dichas capacidades y viceversa.

1.3 ETAPAS FUNDAMENTALES DEL PROCESAMIENTO DE IMAGENES.

El tratamiento digital de imágenes comprende un amplio rango de hardware, software y recursos teóricos. En esta sección se van a presentar las etapas fundamentales que es preciso realizar para el procesamiento de una imagen. La figura 1.1 esquematiza las etapas fundamentales del procesamiento digital de imágenes.

La primera etapa del proceso es la **adquisición** de una imagen digital. Para ello se necesita un sensor de imágenes y la posibilidad de digitalizar la señal producida por el sensor. Este puede ser una cámara de televisión, monocromática o de color, que produce una imagen del dominio del problema cada 1/30 de segundo. El sensor de imágenes puede ser también un digitalizador o escáner que produzca una línea de la imagen cada vez. En este caso, el movimiento de la imagen a través del barrido de líneas produce la imagen bidimensional. Si la salida de la cámara o de otro sensor de imágenes no está todavía en forma digital, puede emplearse un convertidor analógico-digital para digitalizarla. La naturaleza del sensor y la imagen que produce vienen determinadas por la aplicación.

Una vez que se ha obtenido la imagen digital, la siguiente etapa trata del **preprocesamiento** de esa misma. La función básica del preprocesamiento es la de mejorar la imagen de forma que se aumenten las posibilidades de éxito en los procesos posteriores. Como ejemplo de preprocesamiento podemos citar las técnicas de mejorar el contraste, eliminar el ruido y aislar regiones.

La siguiente etapa trata de la **segmentación**. Definida de una forma general, consiste en dividir una imagen de entrada en sus partes constituyentes u objetos. En general, la segmentación no supervisada es una de las labores más difíciles del tratamiento digital de imágenes. Y un algoritmo de segmentación débil o errático casi siempre garantiza que tarde o temprano habrá fallas.

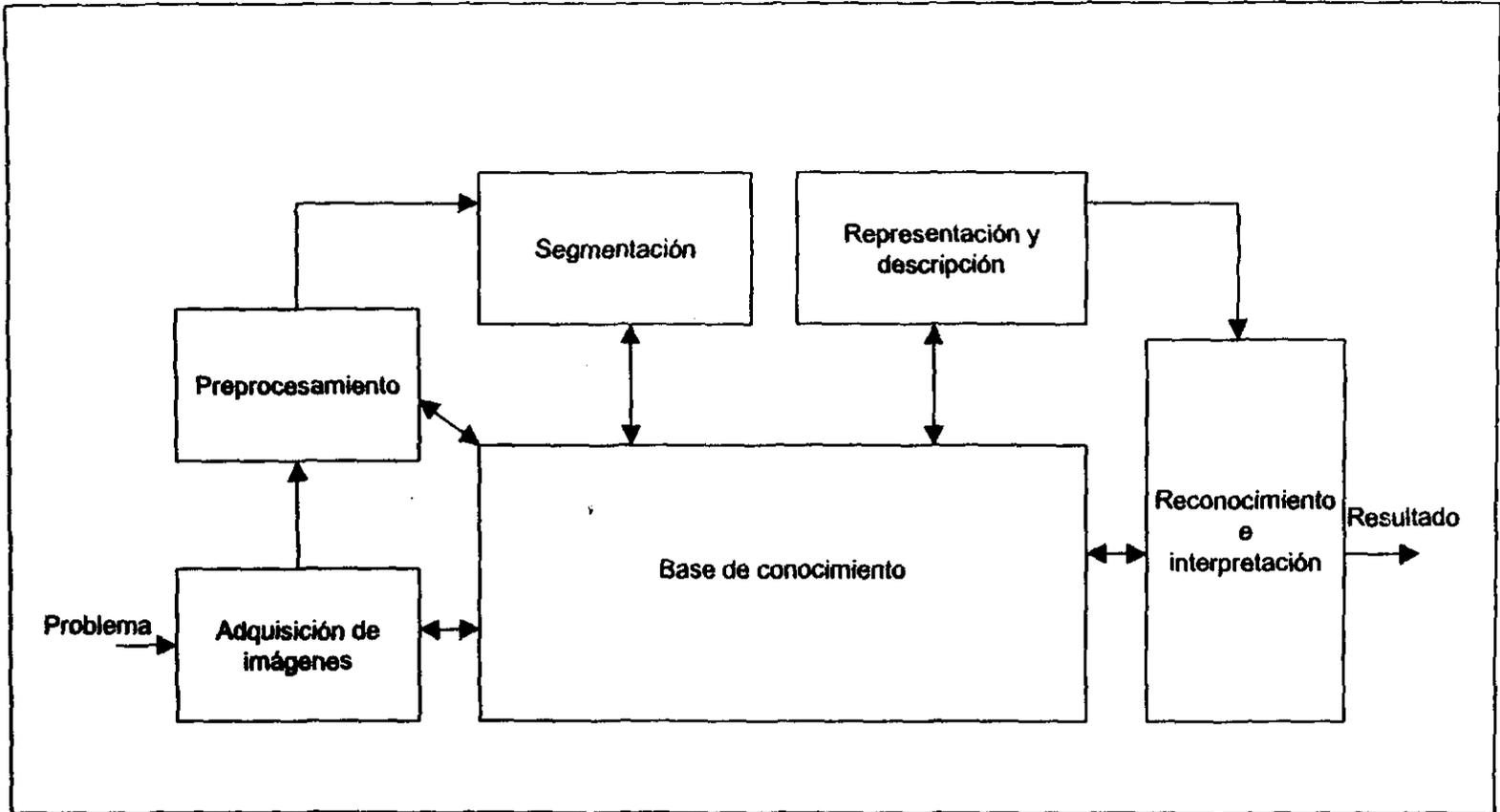


Figura 1.1 Etapas fundamentales del procesamiento digital de imágenes

A la salida del proceso de segmentación habitualmente se tienen los datos que constituyen el contorno de una región o todos los puntos de una región determinada. En cada caso es necesario convertir los datos a una forma adecuada para el procesamiento por computadora. La primera decisión que hay que tomar es si los datos se han de representar como un contorno o como una región completa. La **representación** como un contorno es la adecuada cuando el interés radica en las características de la forma exterior, como las esquinas. Por otro lado, la representación regional resulta adecuada cuando el interés se centra en propiedades internas, como la textura o la estructuración. Sin embargo, en algunas aplicaciones ambas representaciones coexisten. Esto ocurre en las aplicaciones para reconocimiento de caracteres, que a menudo requieren algoritmos basados en la forma de los bordes, así como en la estructuración y otras propiedades internas

La elección de una representación es sólo una parte de la solución para transformar los datos en bruto a una forma adecuada para ser posteriormente tratados por computadora. También debe especificarse un método para describir los datos de forma que se resalten los rasgos de interés.

La **descripción**, también denominada selección de rasgos, consiste en extraerlos con alguna información cuantitativa de interés o que sean fundamentales para diferenciar una clase de objetos de otras.

La última etapa incluye el **reconocimiento e interpretación**. El reconocimiento es el proceso que asigna una etiqueta a un objeto basándose en la información proporcionada por sus descriptores. La interpretación implica asignar significado a un conjunto de objetos reconocidos.

Hasta ahora nada se ha dicho sobre la necesidad del conocimiento previo o sobre la interacción entre la base de conocimiento y los módulos de

procesamiento de imágenes como una base de datos de conocimiento. Este puede ser tan simple como detallar las regiones de una imagen donde se sabe que se ubica información de interés, limitando así la búsqueda que ha de realizarse para hallar tal información. La base de conocimiento también puede ser muy compleja, como una lista interrelacionada de todos los posibles defectos en un problema de inspección de materiales o de una base de datos que contenga imágenes de satélite de alta resolución de una región, en conexión con aplicaciones de detección de cambios. Además de guiar la operación de cada módulo de procesamiento, la base de conocimiento controla también la interacción entre módulos. La comunicación entre módulos de procesamiento se basa generalmente en el conocimiento previo de cómo debe ser el resultado. Este conocimiento no sólo guía la operación de cada módulo, sino que también ayuda en las operaciones de retroalimentación entre módulos a través de la base de conocimiento

Es importante resaltar que la visión de los resultados del procesamiento de imágenes puede realizarse en la salida de cada una de las etapas descritas. También cabe mencionar que no todas las aplicaciones del procesamiento de imágenes necesitan la complejidad de interacciones mencionada. De hecho, en algunos casos ni siquiera son necesarios todos esos módulos. Por ejemplo, el realce de la imagen para interpretaciones visuales humanas rara vez va más allá de la etapa de preprocesamiento. En general, las funciones del procesamiento que incluyen reconocimiento e interpretación están asociadas con las aplicaciones del análisis de imágenes en las que el objetivo es la extracción automática o semi automática de información de una imagen.

1.4 SISTEMA DE PROCESAMIENTO DIGITAL DE IMÁGENES

Los elementos de un sistema de propósito general, capaz de realizar las operaciones de tratamiento digital de imágenes, se representan en la Figura 1.2. Este tipo de sistema generalmente lleva a cabo la adquisición, almacenamiento, procesamiento, comunicación y presentación de imágenes.

1.4.1 Adquisición de imágenes. Para esta parte del proceso se necesitan dos elementos. El primero es un dispositivo físico sensible a una determinada banda del espectro de energía electromagnética (como las bandas de rayos X, ultravioleta, visible o infrarrojo) y que produzca una señal eléctrica de salida proporcional al nivel de energía detectado. El segundo, denominado digitalizador, es un dispositivo para convertir la señal de salida del sistema sensible a forma digital. Como ejemplo, considérese la base de un sistema de imágenes de rayos X. La radiación de una fuente de rayos X se dirige a un objeto, al otro lado del cual se coloca un medio sensible a los rayos X. De esta forma en el medio sensible se forma una imagen de los distintos componentes (como huesos o tejidos) que tienen varios niveles de absorción de rayos X. El propio medio puede ser una película fotográfica, una cámara de televisión combinada con un convertidor de rayos X a fotones, o un sistema discreto de detectores, cuyas salidas estén combinadas para reconstruir una imagen digital.

1.4.2 Almacenamiento. Una imagen de 8 bits y 1024 x 1024 píxeles necesita un millón de bytes de memoria. Así, el proporcionar la capacidad de almacenamiento adecuada suele ser un reto en el diseño de los sistemas de procesamiento de imágenes. El almacenamiento digital para aplicaciones de procesamiento de imágenes cae de lleno en tres categorías básicas: 1) almacenamiento a corto plazo, para ser empleado durante el procesamiento; 2) almacenamiento en línea, para una reutilización relativamente rápida, y 3) almacenamiento en archivo caracterizado por un acceso poco frecuente.

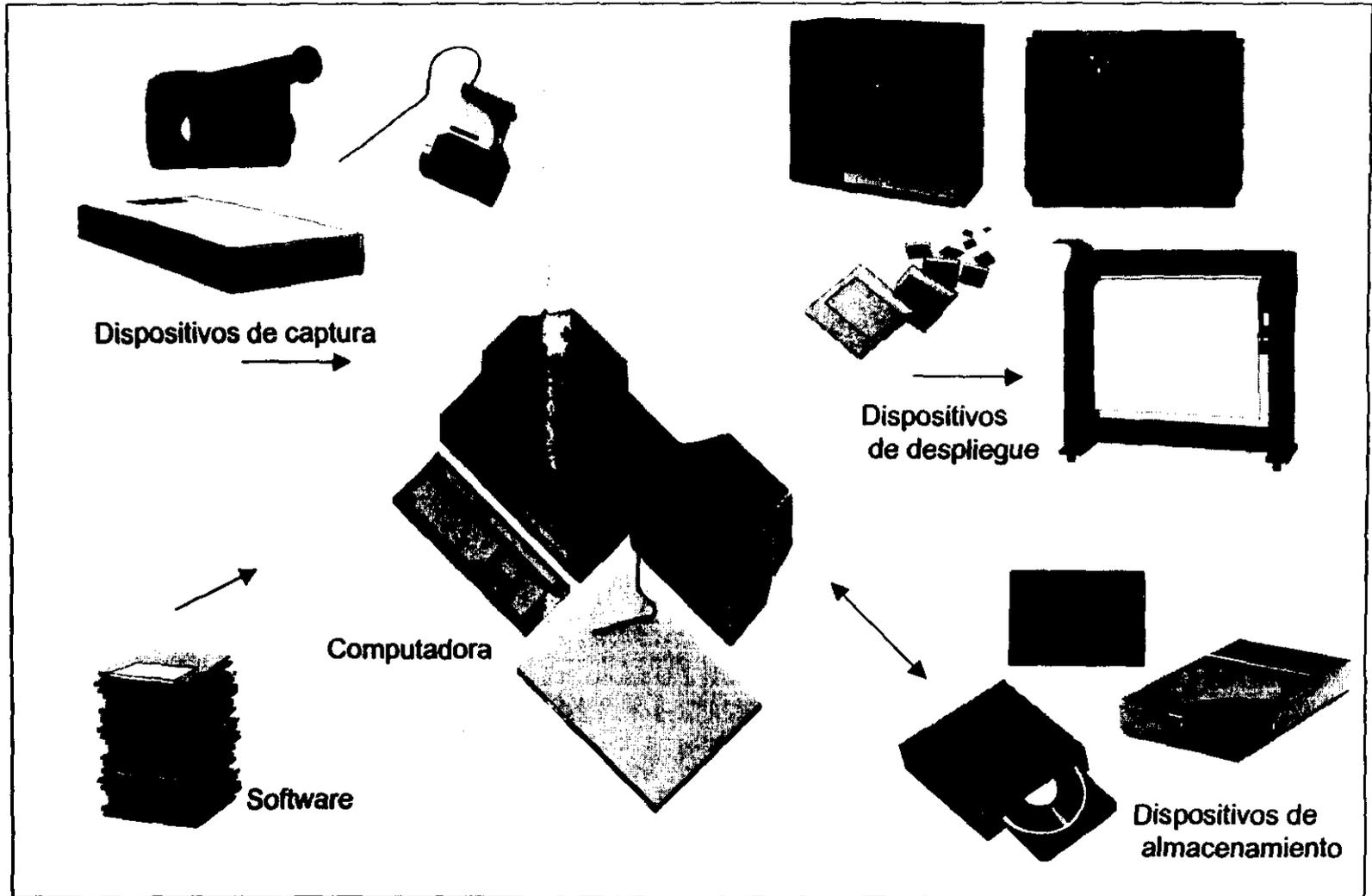


Figura 1.2 Sistema de procesamiento digital de imágenes

Un método para obtener almacenamiento a corto plazo consiste en emplear la memoria de una computadora. Otro, en tarjetas especializadas, denominadas memorias temporales, que almacenan una o más imágenes a las que puede accederse con rapidez, habitualmente a las velocidades de vídeo (30 imágenes completas por segundo). Este último método permite la magnificación (zoom) prácticamente instantánea de la imagen, así como desplazamientos verticales (scroll) y horizontales (pan). Para el almacenamiento en línea generalmente se emplean discos magnéticos. Una tecnología más reciente, denominada almacenamiento magneto-óptico (MO), utiliza un rayo láser y tecnología de materiales especializados para llegar a casi 1 Gbyte de capacidad en un disco óptico de 5 ¼ pulgadas. El factor clave que caracteriza el almacenamiento en línea es el acceso frecuente a los datos. Por ello, las cintas magnéticas así como otros medios de acceso secuencial se utilizan muy poco para almacenamiento en línea para las aplicaciones de procesamiento de imágenes. Finalmente, el almacenamiento en archivo se caracteriza por necesitar un almacenamiento masivo, pero de acceso poco frecuente. Las cintas magnéticas y los discos ópticos son los medios usados habitualmente en las aplicaciones que requieren archivado.

La información anterior se refiere a la capacidad de almacenamiento en forma digital. En aplicaciones en las que no sea necesario recuperar la imagen en forma digital no es raro almacenar la imagen en forma analógica empleando principalmente película fotográfica o cinta de vídeo.

1.4.3 Procesamiento. El tratamiento de imágenes digitales implica procedimientos que normalmente se expresan en forma de algoritmos. Así, con la excepción de la adquisición de las imágenes y su representación, la mayor parte de las funciones de procesamiento de la imagen pueden ser implementadas en software. La única razón de ser del hardware especializado en el procesamiento de imágenes es la necesidad de mayor velocidad en algunas aplicaciones o para evitar algunas limitaciones fundamentales de las computadoras.

Hasta mediados de la década de los ochenta, numerosos modelos de sistemas de procesamiento de imágenes que se comercializaban en todo el mundo eran más bien dispositivos periféricos específicos que se unían a computadoras principales igualmente específicas. Desde finales de la década de los ochenta y en los noventa, el mercado se ha desplazado hacia el hardware de procesamiento de imágenes en forma de placas individuales diseñadas para ser compatibles con las conexiones estándar en la industria y que se pueden instalar dentro de las estaciones de trabajo o en las computadoras personales. Además de rebajar los costos, este desplazamiento del mercado también ha servido de catalizador para muchas nuevas compañías especializadas en el desarrollo de software escrito específicamente para el procesamiento de imágenes.

Aunque todavía se siguen vendiendo sistemas de procesamiento de imágenes para aplicaciones masivas, como el procesamiento de las imágenes de satélites, continúa la tendencia hacia la miniaturización y la de adición de hardware de procesamiento de imágenes a las pequeñas computadoras de propósito general. En particular, el principal hardware de imágenes que ha sido incorporado a estos equipos consiste en una combinación de digitalizador/memoria para la digitalización de la imagen y su almacenamiento temporal, un procesador aritmético/lógico (ALU) para realizar operaciones aritmético-lógicas a la velocidad de los fotogramas y una o varias memorias temporales para el acceso rápido a los datos de la imagen durante el procesamiento. En la actualidad es posible obtener comercialmente una cantidad considerable de software de tratamiento de imágenes. Cuando se combina con otro software de aplicación, como hojas de cálculo y gráficos, proporciona un punto de partida excelente para la solución de problemas específicos de procesamiento de imágenes. Los sofisticados sistemas de presentación junto con el software de tratamiento de textos y generación de informes facilitan la presentación de los resultados. A menudo, las soluciones obtenidas con estos sistemas se pueden transportar posteriormente a tarjetas especializadas capaces

de un rápido procesamiento de las imágenes, y que son compatibles con el equipo usado durante el desarrollo.

El procesamiento de imágenes se caracteriza por soluciones específicas para cada área y en ocasiones para cada problema particular. De tal modo que algunas técnicas que funcionan bien en un área pueden ser totalmente inadecuadas en otra. Lo que ha conseguido la capacidad actual del hardware y del software básico es proporcionar un punto de arranque mucho más avanzado de lo que estaba hace menos de una década a un costo mucho menor.

1.4.4 Comunicación. La comunicación en el tratamiento digital de imágenes implica principalmente, comunicaciones locales entre sistemas de procesamiento de imágenes y comunicaciones remotas entre dos puntos, habitualmente para la transmisión de los datos de las imágenes. El hardware y software para comunicación local que existe en la actualidad es de fácil acceso para la mayoría de las computadoras y las técnicas de compresión y descompresión de imágenes desempeñan un papel principal en esta parte del problema.

1.4.5 Presentación. Los monitores de televisión monocromáticos, y de color, son los principales dispositivos de presentación utilizados en los sistemas actuales de procesamiento de imágenes. Los monitores están gobernados por la salida de una placa de hardware de visualización de imágenes. Las señales de salida del módulo de visualización pueden emplearse como entrada de un módulo de grabación de imágenes que produzca una copia impresa (diapositivas, fotografías o transparencias) de la misma imagen que aparece en la pantalla del monitor.

Los dispositivos de impresión de imágenes son útiles principalmente para el trabajo de procesamiento de baja resolución. Una posibilidad muy simple de generar directamente imágenes en papel con tonos de gris consiste en emplear la capacidad de resaltado de muchas impresoras estándar. El nivel de gris de cada punto impreso se puede controlar a través del número y la densidad de los

caracteres sobreimpresos en este punto. Una adecuada selección del conjunto de caracteres permite alcanzar unas matrices de niveles de gris razonablemente buenas con un programa simple y relativamente pocos caracteres. Otros medios comunes de grabar una imagen directamente sobre papel son las impresoras láser, los dispositivos de impresión térmica y los sistemas de chorro de tinta.

1.5 CONCEPTOS BASICOS.

Cada pixel dentro de la pantalla tiene pixeles que lo circundan, llamados "vecinos". Cada uno de ellos está a una unidad de distancia del pixel seleccionado.

Existen dos formas de tomar en cuenta la vecindad de un pixel con respecto a otro. Un pixel p de coordenadas (x, y) tiene cuatro vecinos verticales cuyas coordenadas vienen dadas por:

$$(x+1, y), (x, y+1), (x, y-1), (x-1, y)$$

Este conjunto de pixeles, denominado los 4-vecinos de p , se representa por $N_4(p)$. Cada pixel está a una unidad de distancia de (x, y) , y algunos de los vecinos de p caen fuera de la pantalla de trabajo si (x, y) está en el borde de la pantalla.

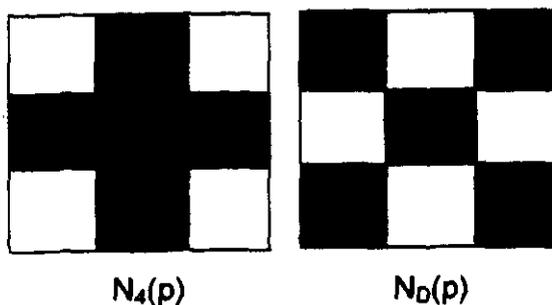


Figura 1.3. 4-vecinos de p .

Los cuatro vecinos en diagonal de p tienen las siguientes coordenadas:

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

y se representan por $N_D(p)$. Estos puntos, junto a los 4-vecinos, se denominan los 8-vecinos de p , y se representan por $N_8(p)$. Al igual que antes, algunos puntos $N_D(p)$ y $N_8(p)$ caen fuera de la pantalla de trabajo si (x, y) está en el borde de la misma.



$N_8(p)$

Figura 1.4. 8-vecinos de p .

El histograma es una herramienta muy importante en el procesamiento digital de imágenes. El de una imagen digital consiste en un diagrama de barras de la propia imagen, utilizándose como abscisas los niveles de gris de la imagen y como ordenadas el número de píxeles de la imagen para cada nivel de color. Es importante mencionar que una imagen específica da lugar a un solo histograma, como parece lógico. Sin embargo, lo contrario no se cumple, es decir, un histograma puede venir de imágenes diferentes.

Una primera ventaja evidente del histograma es que proporciona una descripción global de la imagen. Implica la drástica reducción en la carga computacional de la información existente de una imagen digital, ya que se pasa de una función bidimensional a una función unidimensional.

El realce o mejora de la calidad de una imagen digital, utilizando como herramienta el histograma, se basa en modificar esta gráfica de la imagen original para que, a partir del histograma transformado, obtener una nueva imagen presumiblemente de mayor calidad.

Para entender mejor las posibilidades que ofrece la manipulación del histograma de una imagen es preciso tener en cuenta el tipo de información que contiene. Básicamente, aporta dos parámetros de interés: el rango dinámico de la imagen y su grado de contraste.

Por rango dinámico de una imagen se entiende el rango de todos los posibles niveles de gris que, efectivamente, se encuentran presentes en una determinada imagen. En general, una imagen digital con un rango dinámico reducido presenta una calidad muy pobre, con un elevado nivel de ruido presente. A través de una manipulación del histograma de la imagen original es posible aumentar su rango dinámico y mejorar, consecuentemente, su calidad.

Por contraste de una imagen se entiende la existencia de cambios abruptos de la frecuencia relativa del número de píxeles para diferentes niveles de gris.

El histograma de una imagen digital con un número de colores en el rango $[0, L-1]$ es una función discreta $p(r_k) = n_k / n$, donde r_k es el k -ésimo color, n_k es el número de píxeles de la imagen con ese color, n es el número total de píxeles de la imagen y $k = 0, 1, 2, \dots, L-1$.

De forma general se puede decir que $p(r_k)$ da una idea del valor de la probabilidad de que aparezca el color r_k . Así, la representación gráfica de todos los valores de k presenta una apariencia gráfica de la imagen. El histograma puede usarse como parte de una técnica para mejorar la imagen que se basa sólo en la intensidad de los píxeles individuales.

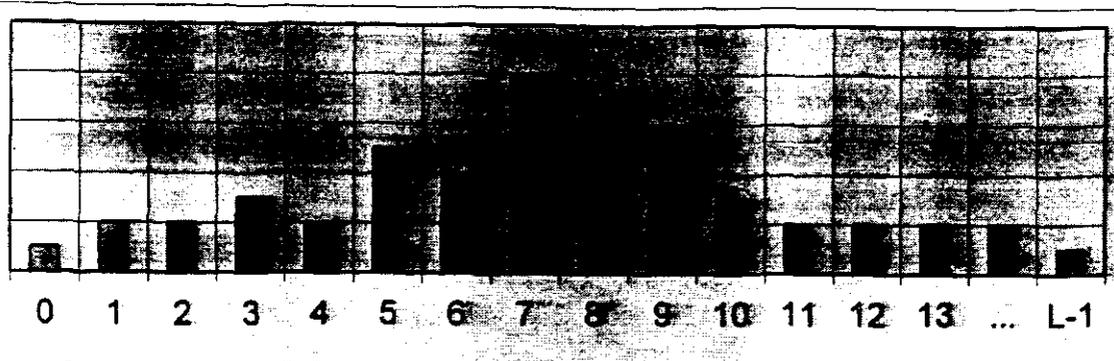


Figura 1.5. Histograma de una imagen digital.

En la figura 1.5 se puede apreciar que la mayor cantidad de colores se agrupan al centro de ésta, lo que da una idea de los colores predominantes en la imagen. Cabe mencionar que existen algoritmos para la ecualización de un histograma. Estos algoritmos sirven para cambiar el número de píxeles de cada color y la forma en que éstos se reparten y agrupan dentro de la imagen; la idea es que el histograma se logre convertir a un histograma plano, donde el número de píxeles de un determinado color sea aproximadamente igual a los demás.

En la figura 1.6 se muestran los histogramas correspondientes a cuatro tipos básicos de imágenes: una imagen oscura y una brillante; una imagen de bajo contraste y una de alto contraste.

Las transformadas lineales, especialmente las de Fourier y Laplace, son bien conocidas por proporcionar técnicas para resolver problemas en sistemas lineales. Característicamente se usa una transformación como una herramienta física o matemática para cambiar el problema a uno que pueda resolverse. La representación de una función en términos de funciones periódicas sencillas, tales como seno y coseno, tienen una gran importancia práctica.

1.6 SEGMENTACIÓN

En la *compresión o realce de imágenes*, lo que se busca es una aproximación a, o una versión mejorada de la imagen de entrada. Mas en el caso de análisis de imágenes, la entrada es igualmente una imagen, pero se desea como salida una descripción de la misma. En realidad, su descripción se refiere básicamente a las partes que la componen, para generar esta descripción es necesario *segmentar o dividir la imagen en estas partes*.

Algunas operaciones de segmentación pueden aplicarse directamente a cualquier imagen. Otras solo pueden aplicarse a una imagen que ya haya sido segmentada parcialmente, ya que dependen de la geometría de las partes que ya han sido extraídas.

Haralick y Shapiro (1992) definen una segmentación como la partición de una imagen en un conjunto de regiones que no se traslapan y cuya unión es la imagen entera. Consideran que debe cumplir las siguientes características: las regiones de una segmentación deben ser uniformes y homogéneas respecto a ciertas características tales como el nivel de gris o la textura; el interior de la región debe ser sencillo y sin muchos pequeños agujeros; las regiones adyacentes de una segmentación deben tener valores significativamente diferentes respecto a las características en las cuales son uniformes; y finalmente las fronteras de cada segmento deben ser simples, lisas y espacialmente exactas. Sin embargo, lograr estas propiedades deseadas es sumamente difícil, ya que las regiones estrictamente uniformes y homogéneas están típicamente llenas de pequeños agujeros y tienen bordes dentados. El hecho de insistir que las regiones adyacentes tengan significativas diferencias de valor puede dar como resultado que las regiones se fundan y se pierdan las fronteras.

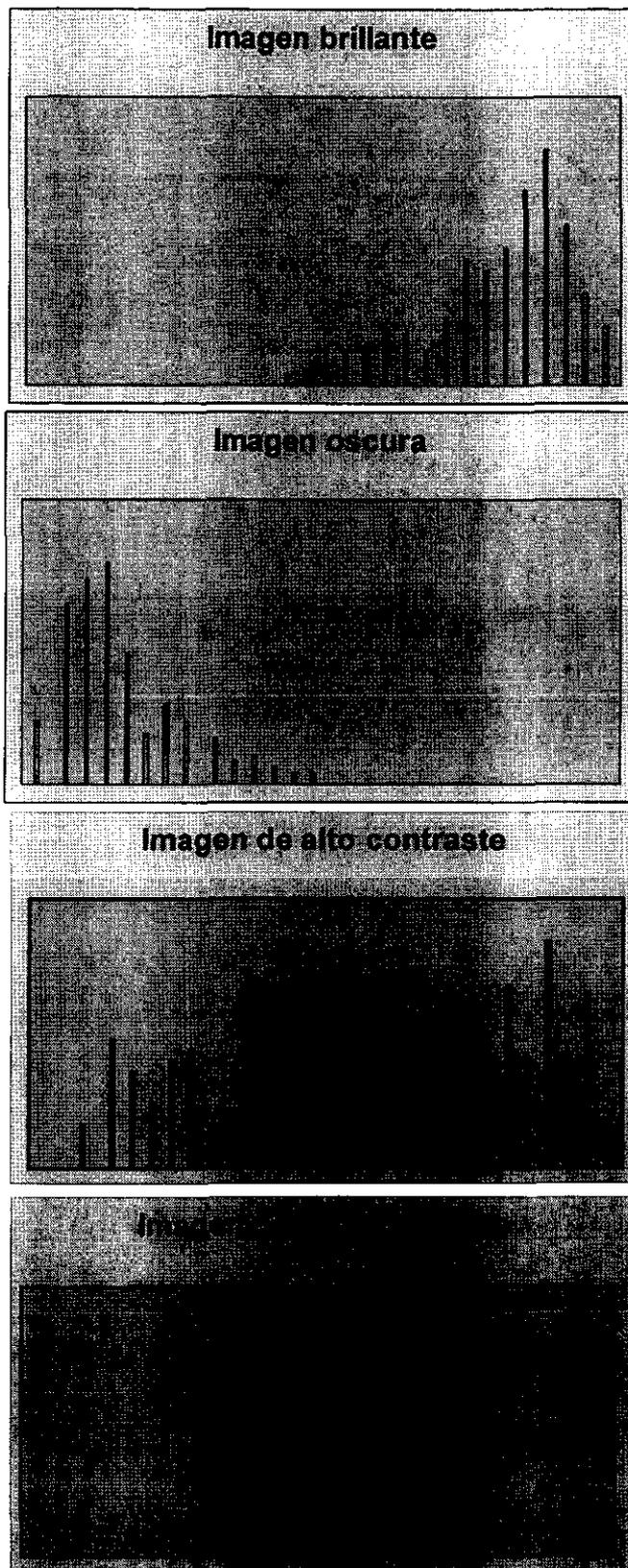


Figura 1.6 Histogramas de una imagen brillante, una oscura, una de alto contraste y una de bajo contraste

Es importante enfatizar que no hay una aproximación única a la segmentación. Muchos tipos diferentes de partes de una imagen pueden servir como los segmentos en los que se basa la descripción. También hay muchas formas en que se puede intentar extraer partes de una imagen. Aún no se entienden bien los procesos de la percepción visual humana involucrados en la segmentación de una imagen. Por tanto, no hay un criterio único para juzgar una segmentación exitosa. Debe juzgarse de acuerdo a la utilidad de la descripción que se obtiene usando los objetos resultantes.

El primer paso del análisis de imágenes consiste generalmente en segmentarla. La segmentación subdivide una imagen en sus partes constituyentes u objetos. El nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. Esto es, la segmentación deberá detenerse cuando los objetos de interés de una aplicación hayan sido aislados o reconocidos; por ejemplo, en las aplicaciones autónomas de adquisición de blancos aire tierra el interés tiende, entre otras cosas, a la identificación de los vehículos de una carretera. El primer paso consiste en segmentar la carretera de la imagen y a continuación segmentar del contenido de la carretera los objetos cuya longitud o tamaño correspondan a la de los potenciales vehículos. No tiene objeto llevar la segmentación por debajo de esta escala, ni tampoco hay ninguna necesidad de intentar segmentar componentes de la imagen situados fuera de los límites de la carretera.

En general, la segmentación autónoma es una de las tareas más difíciles del procesamiento de imágenes. Esta etapa del proceso determina el eventual éxito o fracaso del análisis.

De hecho, la segmentación rara vez llega a alcanzar una solución satisfactoria. Por esta razón, se debe poner un considerable cuidado en aumentar la probabilidad de tener una segmentación robusta. En algunas situaciones, tales como las aplicaciones de inspección industrial, es posible que a veces se tome

alguna medida de control sobre el entorno. Los diseñadores de sistemas de procesamiento de imágenes con experiencia prestan invariablemente una considerable atención a oportunidades de este tipo. En otras aplicaciones, como la adquisición de objetivos, el diseñador del sistema no tiene control sobre el entorno. Por consiguiente, el método habitual se orienta a la selección del tipo de sensor más apropiado para realzar los objetos de interés mientras se disminuye la contribución de los componentes irrelevantes de la imagen. Un buen ejemplo es la utilización de imágenes infrarrojas para detectar objetos con una firma fuerte de calor tales como carros de combate en movimiento.

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las propiedades básicas de los valores del nivel de gris: discontinuidad y similitud. En la primera categoría, el método consiste en dividir una imagen basándose en los cambios bruscos de nivel de gris. Las principales áreas de interés de esta categoría son la detección de puntos aislados y la detección de líneas y bordes de una imagen. Los principales métodos de la segunda categoría están basados en la amplitud del umbral, conocida como umbralización, el crecimiento de región basado en la discontinuidad o similitud de los valores del nivel de gris de sus píxeles es aplicable tanto a las imágenes estáticas como a las dinámicas (variantes en el tiempo). En el último caso, sin embargo, el movimiento puede utilizarse a menudo como un potente indicador para mejorar el rendimiento de los algoritmos de segmentación.

1.7 TÉCNICAS DE SEGMENTACIÓN

a) Agrupamiento espacial guiado por la medida espacial. Básicamente lo que se realiza en este tipo de segmentación es calcular el histograma de la imagen y buscar si en el histograma hay valles. Estos valles son la división entre los dos o más cúmulos y se considera que las colinas del histograma corresponden a diferentes objetos en la imagen. El caso más sencillo de estos métodos es el del histograma bimodal en el que se tiene un objeto brillante contra un fondo oscuro

y solamente es necesario separar el fondo del objeto. Sin embargo no todas las imágenes tienen estas características e incluso no siempre se encuentran valles claramente definidos en el histograma.

b) Crecimiento de regiones. Este grupo de métodos tiene como base elegir varios píxeles como nodos de una gráfica. Para cada uno de estos nodos, se verifica si sus vecinos son suficientemente parecidos en propiedades, en cuyo caso deben unirse por un arco. Los segmentos de imágenes son juegos de píxeles máximos que pertenecen todos al mismo componente conectado. Para estos métodos por lo general toman los 4-vecinos del píxel. Esta familia de métodos tiene la indiscutible ventaja de su simplicidad. Sin embargo si uno de los arcos es tomado incorrectamente esto puede causar que se unan dos regiones vecinas que no deberían unirse. Para corregir este problema se usa un instrumento más poderoso que puede ser un operador de gradiente para detectar las fronteras y con base a esto decidir. También se han usado los centroides de regiones predefinidas para verificar que un píxel realmente pertenece a esta región.

c) Combinaciones de ligado híbridas. Las dos soluciones anteriores: usar un operador y usar un centroide pueden combinarse de modo que se aprovechen sus fuerzas relativas. La fuerza de un ligado simple es que las fronteras se colocan de una manera especialmente precisa. Su debilidad es que los espacios de las orillas dan como resultado fusión excesiva. La fortaleza del ligado por centroide es su habilidad para colocar fronteras en áreas con un gradiente débil. Esto es posible porque no depende de una gran diferencia entre el píxel y su vecino para declarar una frontera. Depende de una gran diferencia entre el píxel y la media de la región vecina para declarar una frontera. Lo que hace esta técnica es claro, el ligado de centroide se hace sólo para los píxeles que no están en la orilla, o sea el crecimiento de la región no se permite a través de los píxeles de la orilla, de tal modo que si los parámetros del ligado por centroide fueran fijados para una diferencia aunque sea grande, entre el valor del píxel y la media de la región considerada lo suficientemente pequeña para permitir la unión. Por otra parte, la

técnica de combinación híbrida de dos pasos trabaja los componentes conectados de los píxeles que no están en los bordes.

d) Agrupamiento espacial. Es posible determinar las divisiones de la imagen combinando el agrupamiento espacial con el crecimiento de regiones. A esto se le llama agrupamiento espacial. Básicamente, los esquemas de este tipo de agrupamiento combinan la técnica de búsqueda de modo del histograma con el *crecimiento espacial* o con una *técnica de ligado espacial*. Lo que se hace en este caso es localizar los picos del histograma y a partir de ellos determinar la localización de todos los píxeles que corresponden a la medida del pico. Luego empezando con un píxel que corresponda al pico más alto no procesado, se busca si sus vecinos pueden ser etiquetados como pertenecientes a la misma región.

e) División y unión. Esta serie de métodos considera que toda la imagen es una sola región. Sucesivamente se va dividiendo en cuartos si el segmento no es suficientemente homogéneo, esto es si la diferencia entre el mayor y el menor nivel de gris es grande. Un método de unión empieza con una segmentación inicial y sucesivamente une las regiones suficientemente similares. Debido a que los segmentos son sucesivamente divididos en cuartos, las fronteras producidas por esta técnica tienden a ser cuadrangulares y un poco artificiales. A veces algunos segmentos provenientes de segmentos adyacentes divididos deberían ser unidos en lugar de quedar separados.

f) Segmentación basada en reglas. Las reglas detrás de cada uno de los métodos comentados anteriormente se encuentran codificados en los procedimientos del método. Por lo tanto, no es fácil tratar aproximaciones diferentes sin volver a programar. Esto puede solucionarse por medio de un sistema experto basado en las reglas para la segmentación. El conocimiento en el sistema no es un dominio específico de aplicación pero incluye conocimiento de propósito general independiente de la escena acerca de imágenes y criterios de agrupamiento.

g) Segmentación basada en movimiento. Si se dispone de una cámara fija que toma una escena en movimiento, los datos son una secuencia de imágenes en lugar de una sola imagen. En este caso, los objetos se ven desde diferentes posiciones, por lo que el movimiento de los mismos crea un cambio en las imágenes que puede ser muy útil para ayudar a localizarlos. Los procesos de segmentación deben enfocarse a los objetos que difieren entre imágenes.

Las técnicas de segmentación son básicamente técnicas *ad hoc* que difieren precisamente en la forma en que subrayan una o más de las propiedades deseadas y en que logran una solución "de compromiso" entre una propiedad y otra, por lo que cada uno de estos métodos es adecuado para algunas imágenes e inadecuado para otras.

CAPITULO II: MARCO TEORICO: SEGMENTACION BASADA EN TEXTURA.

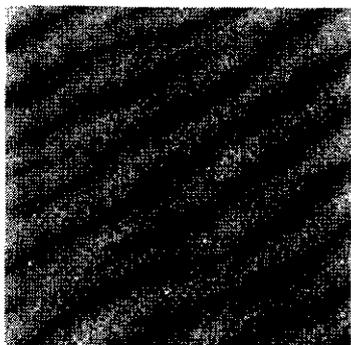
Para poder realizar una segmentación de este tipo para una imagen es necesario ser capaces de discriminar texturas. Por ello, se describe qué es una textura y posteriormente algunas de las familias de métodos de caracterización o reconocimiento de las mismas. Finalmente, es posible hablar de una segmentación basada en estas.

Una región es un grupo de píxeles máximamente conectados que poseen la propiedad de un cierto nivel de gris. La región de nivel de gris puede ser evaluada respecto a su área y forma. El nivel primitivo de gris incluye tanto su nivel de gris como las propiedades de la región del nivel de gris. La organización espacial puede ser aleatoria, puede tener una dependencia por pares de un primitivo sobre un primitivo adyacente, o bien puede tener una dependencia de n primitivos a la vez.

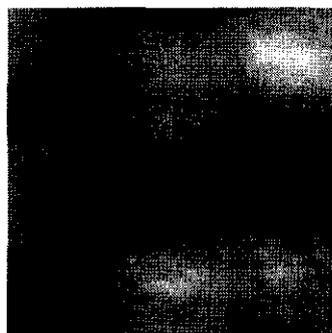
2.1 TEXTURAS.

En una imagen, un área con textura se caracteriza por una distribución espacial no uniforme de intensidades de gris. A pesar de que las imágenes a color también tienen esta distribución, para efectos de este trabajo sólo se tomarán en cuenta los métodos para manejo de texturas en imágenes de tonos de gris. La figura 2.1 presenta varios ejemplos.

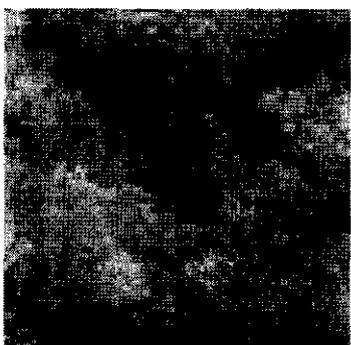
Cuando una zona de una imagen tiene poca variación de los niveles de gris primitivos la propiedad dominante de esa área es el nivel de gris. Cuando una zona tiene una gran variación de los niveles primitivos de gris, la propiedad dominante es la textura.



a)



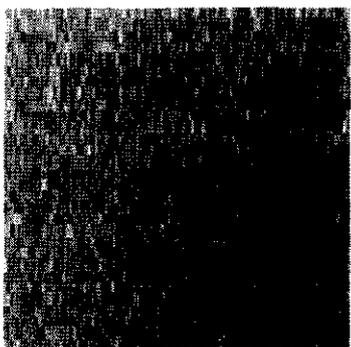
b)



c)



d)



e)



f)

Figura 2.1 Ejemplos de diferentes texturas. a) Arena. b) Nubes. c) Árboles. d) Gotas de agua sobre una superficie plana. e) Granito. f) Rocas

Obviamente, la textura no es una propiedad local, sino regional. No se puede hablar de que un pixel en particular tiene una textura, sino que una región la tiene. Así pues, un pixel con una cierta intensidad de gris podría pertenecer a varios objetos de los que componen una imagen. En el caso de la figura 2.2 un pixel de un gris medio podría pertenecer a las plantas del fondo, ser parte del pasto de la parte del frente o incluso ser parte de alguna de las rocas que se encuentran en la barda. De hecho, aún una región pequeña de pixeles podría cumplir con lo descrito anteriormente.

En general, en casi cualquier imagen existe cierta textura, pero en ocasiones ésta no es tan marcada que pueda usarse para realizar una segmentación adecuada. Sin embargo, en otros casos es necesario que la segmentación sea basada en la textura. La segmentación basada en la textura es posible aún cuando no son aparentes los bordes con intensidades diferentes entre las diversas regiones. En la figura 2.2 la intensidad de las plantas y la de las rocas en la parte del frente es similar, pero sus texturas difieren.

La variación en la intensidad que caracteriza a una textura ordinariamente refleja alguna variación física en la escena que representa. Se puede considerar que una textura se compone de piezas: el tamaño, la forma, los tonos y el arreglo espacial de las regiones son factores críticos para diferenciar diversas texturas. Las regiones con una textura pueden ser de hecho la partición de la imagen como es el caso de los árboles en la figura 2.1 c) o estar distribuidas en un fondo homogéneo, como las gotas de agua en la figura 2.1 d).

La textura es una característica importante para el análisis de muchos tipos de imágenes. Algunos ejemplos del tipo de profesionales que usan las texturas son los que trabajan en procesamiento remoto, ellos analizan texturas en las imágenes multiespectrales obtenidas desde aviones, o satélites, los biomédicos analizan las texturas de imágenes microscópicas de cultivos celulares o muestras de tejidos y quienes se dedican a visión artificial las utilizan para analizar las

escenas al aire libre y las superficies de objetos. A pesar de su importancia y de su frecuencia de aparición en las imágenes, no hay una definición precisa de textura. Las técnicas que discriminan las texturas son mayormente *ad hoc*. Lo que se hace para trabajar con texturas es que, dada una imagen con muchas áreas texturizadas, se determinan las fronteras entre sus diferentes regiones. Para ello, por lo general es necesario dada una textura definir un modelo que la describa y teniendo un modelo que describa las diferentes texturas de la imagen determinar a cuál de ellas pertenece cada región.



Figura 2.2 Imagen con varias texturas.

Una imagen con textura se describe por el número y tipo de sus niveles primitivos de gris y por la organización espacial o distribución de estos. Los niveles

primitivos se definen como regiones con propiedades específicas de nivel de gris. El nivel primitivo de gris puede ser descrito como el nivel promedio, o como los niveles máximo y mínimo de su región.

Cualitativamente, se puede decir que la textura de una imagen tiene una o más de las siguientes propiedades: finura, grosor, blandura, granulación, aleatoriedad, alineación o ser moteada, irregular, o rugosa. Cada una de estas cualidades se traduce en alguna propiedad de nivel primitivo de gris y en la interacción espacial entre ellas. Desafortunadamente, no es trivial el problema de mapear el significado semántico representándolo como propiedades precisas de los niveles primitivos de gris y su distribución espacial. Sin embargo, se puede decir que cuando el patrón espacial de los niveles primitivos de gris es aleatorio y la variación entre los niveles primitivos de gris es amplia, se tiene como resultado una textura fina. Cuando el patrón espacial se vuelve más definido y las regiones de nivel de gris abarcan más y más píxeles se tiene una textura más gruesa.

El advenimiento de análisis automático de imágenes dio como resultado dos aproximaciones principales al análisis de texturas: la aproximación estadística y la aproximación estructural. La aproximación estadística genera parámetros para caracterizar las propiedades estocásticas de la distribución espacial de niveles de gris en una imagen. La aproximación estructural analiza escenas visuales en términos de la organización y relaciones entre sus subestructuras.

Los primeros trabajos en análisis de texturas buscaron descubrir objetos útiles que tenían alguna relación con la fineza y grosor, contraste, direccionalidad, rugosidad y regularidad de la textura de una imagen. Estas aproximaciones de la textura de objetos incluyeron el uso de una función de autocorrelación, la cantidad de bordes por unidad de área, la probabilidad espacial de co-ocurrencia de niveles de gris, las distribuciones espaciales de extremos relativos y la morfología matemática.

Ciertos métodos que se desarrollaron posteriormente para realizar el análisis de textura en imágenes buscaron un entendimiento más profundo de lo que es la textura de una imagen, y por esta razón usaron modelos generativos de imágenes. Algunas de las técnicas basadas en modelos de series de tiempo con promedio móvil autorregresivos, campos aleatorios Markovianos, energía de las texturas y modelos de mosaicos aleatorios caen dentro de esta categoría.

Estos métodos para analizar las texturas se describen a continuación, de acuerdo con la cronología de su primera aparición. Posteriormente se describe la forma de utilizarlos para realizar la segmentación basada en texturas.

2.2 CO-OCURRENCIA DE NIVELES DE GRIS.

A principios de los sesenta surgió el primero de un grupo de métodos que utilizan la relación de la matriz de niveles de gris consigo misma, pero desplazada cierta distancia y girada un cierto ángulo, que puede ser por ejemplo 0° , 45° , 90° ó 135° . La esencia de este método consiste en que caracteriza a una textura por la probabilidad de que ocurra cierta relación de píxeles a una distancia dada, a un ángulo dado.

La solución de la dependencia espacial del nivel de gris caracteriza a la textura por la co-ocurrencia de sus niveles de gris. Las texturas gruesas son aquellas para las cuales la distribución cambia sólo ligeramente con la distancia, y las texturas finas aquellas para las cuales los cambios de distribución son grandes para la distancia.

El nivel de co-ocurrencia de gris puede ser especificado en una matriz de frecuencias relativas en la cual existen en la imagen dos píxeles vecinos separados por la distancia dada, uno con el nivel de gris correspondiente a la columna y el otro con el nivel de gris de la fila. Para perfeccionar estos modelos lo que se ha hecho es calcular la co-ocurrencia de gris relacionado sólo con los

pixeles cerca de los bordes, o usar estadísticas de co-ocurrencia generalizada y finalmente se suele combinar la co-ocurrencia de niveles de gris y la co-ocurrencia en los bordes. Un aspecto importante de la textura que puede ser capturado de las matrices de co-ocurrencia es la direccionalidad, esto es, la diferencia en la correlación de los niveles de gris como función de la dirección.

La fortaleza de este modelo de la co-ocurrencia de niveles de gris es que caracteriza la interrelación espacial de los niveles de gris en un patrón de textura y puede hacerlo de modo que es invariable bajo transformaciones de nivel de gris. Su debilidad es que no captura los aspectos de forma de los niveles primitivos del gris. Por lo tanto, no trabaja bien para texturas compuestas de áreas grandes. Tampoco puede capturar las relaciones espaciales entre los niveles de gris primitivos cuyas regiones son mayores que un pixel.

Los objetos de textura pueden ser calculados a partir de una matriz de co-ocurrencia de niveles espaciales de gris. La matriz de co-ocurrencia puede ser normalizada dividiendo cada elemento entre la suma de todos los elementos de la matriz. También es posible calcular matrices de probabilidad condicional para objetos de textura, con la ventaja de que no les afectan los cambios en el histograma de niveles de gris dentro de la imagen.

Un conjunto de descriptores útiles para el análisis de texturas por autocorrelación incluye uniformidad de energía, entropía, probabilidad máxima, contraste, correlación, probabilidad de una corrida larga, homogeneidad y tendencia a agrupamiento.

En la figura 2.3 se muestra una matriz 4 x 4 que se trata como la imagen de entrada y los cálculos correspondientes a la co-ocurrencia de esta imagen. Se ha tomado una distancia de 1 y ángulos de 0, 45, 90 y 135 grados, y solamente cuatro niveles de gris posibles, para la mejor comprensión del método.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

a)

		Nivel de Gris			
		0	1	2	3
Nivel de Gri	0	#(0, 0)	#(0, 1)	#(0, 2)	#(0, 3)
	1	#(1, 0)	#(1, 1)	#(1, 2)	#(1, 3)
	2	#(2, 0)	#(2, 1)	#(2, 2)	#(2, 3)
	3	#(3, 0)	#(3, 1)	#(3, 2)	#(3, 3)

b)

0°	$P_H =$	4	2	1	0	90°	$P_V =$	6	0	2	0
		2	4	0	0			0	4	2	0
		1	0	6	1			2	2	2	2
		0	1	1	2			0	0	2	0
135°	$P_{LD} =$	2	1	3	0	45°	$P_{SD} =$	4	1	0	0
		1	2	1	0			1	2	2	0
		3	1	0	2			0	2	4	1
		0	0	2	0			0	0	1	0

c)

Figura 2.3 a) Valor de los pixeles de la "imagen" original. b) Matriz que muestra cómo se calcula la coocurrencia para cada uno de sus elementos. c) Cálculo de la coocurrencia para 0°, 45°, 90° y 135°.

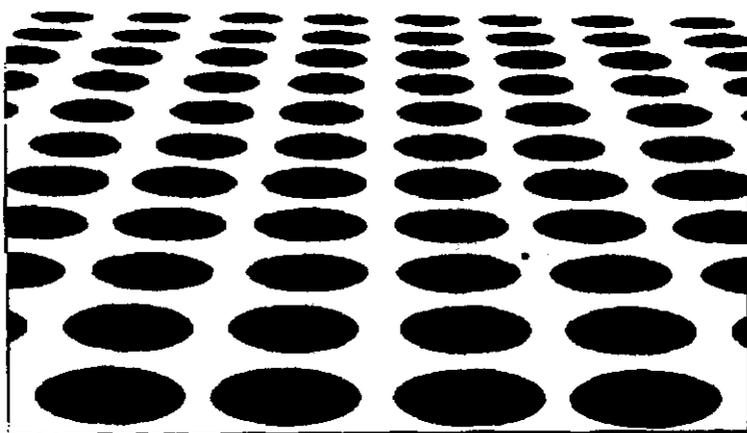


Figura 2.4 Imagen típicamente usada para calcular la forma a partir de la textura. De este tipo de imágenes se puede inferir la inclinación del plano.

2.3 FORMA A PARTIR DE LA TEXTURA.

Los gradientes de textura de las imágenes en fotografías oblicuas pueden usarse para estimar la orientación de la superficie del objeto tridimensional observado. Las técnicas desarrolladas hasta el momento suponen que la textura observada no tiene cambios de profundidad y no hay cambios de textura dentro del área observada. Tampoco toman en cuenta la posibilidad de subtexturas.

A principios de los sesenta se produjo el diseño conceptual de un sistema llamado VISILOG, el cual podía dirigir un vehículo que se movía libremente a través de un ambiente indeterminado. Una clase importante de información necesaria para guiar un vehículo es la orientación de la superficie sobre la cual se mueve éste. La base del diseño fue un análisis que relacionaba la superficie con el gradiente de textura en proyección con la perspectiva de la imagen. Se asumió que una superficie estocástica regular se observa a través de proyección en perspectiva y el número de elementos de textura pueden ser medidos a lo largo de dos segmentos de línea paralelos perpendiculares a la dirección de visión. Se mide el número de elementos en una línea, midiendo el número de cambios en brillo a lo largo de la línea. El número de cambios en brillo es el número de extremos relativos.

Esta línea de investigación se ha seguido trabajando. Blostein y Ahuja (1997) proponen que si el área de las primitivas de textura es constante, entonces no es necesario que la densidad de las primitivas de textura sea constante. Lo que ellos desarrollan es una relación entre el área de una primitiva en el plano de proyección en perspectiva con la inclinación de profundidad y la inclinación lateral de la superficie del plano que tiene la textura. Cualitativamente, la forma a partir de la textura puede funcionar; sin embargo, hasta el momento las técnicas cuantitativas no se han perfeccionado. En la figura 2.4 puede verse el tipo de textura que se reconoce con este tipo de métodos.

2.4 MODELOS DE MOSAICOS ALEATORIOS.

También hacia finales de los sesenta surgió la idea de usar mosaicos aleatorios. Los modelos de este tipo se construyen en dos etapas. La primera, utiliza algún medio para dividir el plano en células, y la segunda, asigna un valor propio a cada una de las células en las que se dividió el plano en la etapa anterior.

En una primera parte, el plano es dividido por un conjunto aleatorio de líneas. Cada célula es una región conectada cuya frontera consiste en los segmentos de recta de las líneas aleatorias. Es decir, se divide el plano en células convexas que constan de segmentos de recta del conjunto de líneas generadas aleatoriamente. Por medio de este modelo se obtienen células triangulares. Otro modelo de mosaico aleatorio hace una división basada en el proceso de intensidad λ de Poisson, que siembra los puntos en el plano. Cada punto determina una célula que consta de todos los puntos en el plano que están más cercanos a cada uno de los puntos sembrados anteriormente. Este modelo lleva a regiones que tienen forma hexagonal. Otro modelo que pertenece a esta familia se basa en el anterior. Se siembran puntos y se divide como en el modelo anterior, además se traza una línea entre cada par de puntos sembrados, cuyas células correspondientes comparten un segmento común de frontera. En este caso las células tienden a ser cuadriláteros.

A principios de los ochenta Modestino, Fries y Vickers (1980) propusieron un método que se basa en calcular la función de densidad del poder espectral para un plano segmentado por un proceso de línea aleatoria en el cual los niveles de gris de una célula tiene una dependencia markoviana sobre los niveles de gris de las células que los rodean. Incluso se intentó sobreponer un campo markoviano aleatorio para describir las transiciones entre las células.

2.5 BORDES DE LAS TEXTURAS

A principios de los setenta surgió una nueva solución, totalmente diferente de las anteriores. Esta serie de métodos conciben la textura no en términos de frecuencia espacial sino en términos de bordes por unidad de área. Un borde que pasa a través de un pixel puede ser detectado comparando los valores de las propiedades locales obtenidas en pares de vecindarios no traslapados que tienen frontera con el pixel. Para detectar los microbordes deben usarse pocos vecinos. Para detectar macrobordes se usan muchos vecinos.

Los bordes pueden ser detectados por medio de una variedad de operadores matemáticos locales que esencialmente miden alguna propiedad relacionada con el gradiente de la función de intensidad de la imagen, por lo que a estos métodos también se les conoce como de análisis de gradiente.

Davis y Mitiche (1980) y Mitiche y Davis (1981) proponen un procedimiento que implica aplicar un operador sensible a los bordes, dividir los resultados de acuerdo al operador de bordes y finalmente calcular picos de los puntos que están sobre el nivel de división.

Por su parte, Kotanzad y Chen (1989) sugieren un método que no requiere información *a priori*. Se basa en detectar cambios en las características de la textura de pequeñas regiones locales.

2.6 TRANSFORMACIONES ORTOGONALES

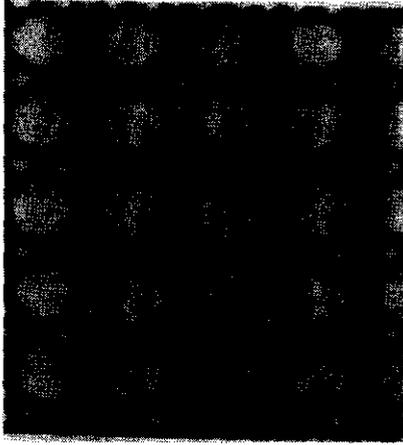
También hacia principios de los setenta surgió la primera solución con base en una transformada digital. Las características de la frecuencia espacial de imágenes bidimensionales pueden ser expresada por la función de autocorrelación o por el espectro de potencias de dichas imágenes. Ambos pueden ser calculados digitalmente o implementados en un sistema óptico de tiempo real. En general, los

objetos basados en el espectro de potencias de Fourier han mostrado desempeñarse más pobremente que los objetos basados en estadísticas de co-ocurrencia de niveles de gris de segundo orden. Incluso tienen peor desempeño que aquellas basadas en estadísticas de primer orden de diferencias espaciales de niveles de gris aunque las estadísticas expresan que este efecto, si en realidad existe, es mínimo. Otras transformadas, además de la de Fourier, pueden ser usadas para el análisis de texturas.

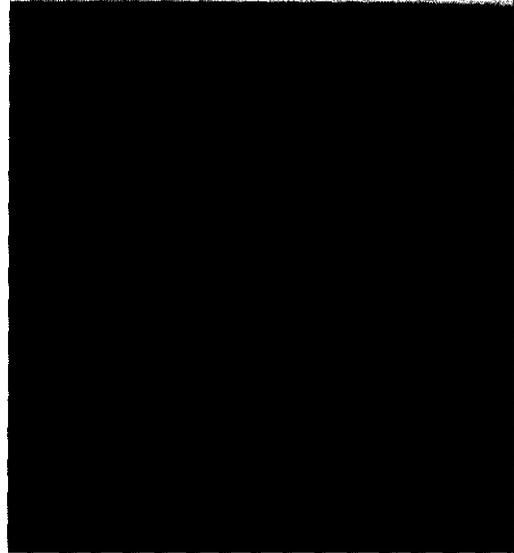
En el método de análisis de textura de transformación digital, la imagen digital se divide en un cierto número de subimágenes de pequeños cuadrados que no se traslapan. Bajo el supuesto de que el tamaño de la subimagen está en células de resolución de $n \times n$; entonces los niveles de gris n^2 pueden ser concebidos como los componentes n^2 de un vector dimensional n^2 . El grupo de subimágenes constituye entonces un grupo de vectores dimensionales n^2 . En la técnica de transformación cada uno de estos vectores es reexpresado en un nuevo sistema coordinado.

El objeto de la transformación es que los vectores básicos del nuevo sistema coordinado tienen una interpretación relacionada con la frecuencia espacial o secuencia, y como la frecuencia está estrechamente relacionada a la textura, tales transformaciones pueden ser útiles.

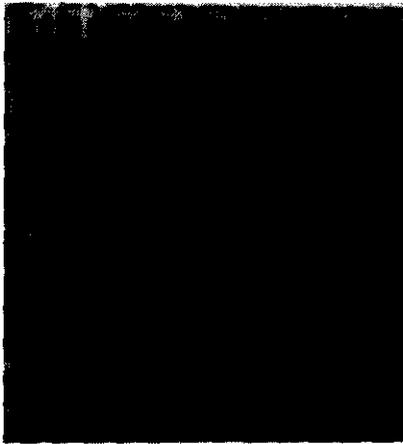
El nivel primitivo de gris en los modelos de frecuencia espacial (secuencia) es el nivel de gris. La organización espacial se caracteriza por la clase de dependencia lineal que mide el largo de proyección. Esta aproximación propone un modelo de síntesis de texturas donde un arreglo estocástico de textura se produce aplicando algún operador espacial a un arreglo de variables aleatorias independientemente distribuidas idénticamente. Lo que sugieren es que para analizar una textura se debe tratar de eliminar la correlación de los niveles de gris.



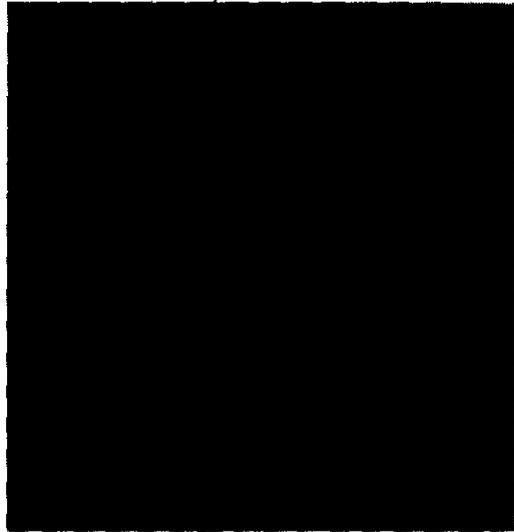
a)



b)



c)



d)

Figura 2.5. a) Imagen con una textura periódica. b) Espectro de la imagen. c) Otra imagen con diferente textura. d) Espectro de la imagen c.

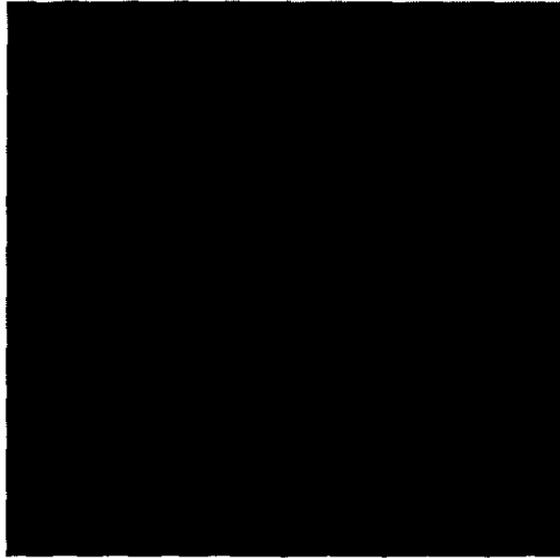
El espectro de Fourier está idealmente indicado para describir la direccionalidad de patrones bidimensionales periódicos o casi periódicos de una imagen. Estos patrones de textura global, aunque son fácilmente distinguibles como concentraciones de alta energía del espectro, generalmente son bastante difíciles de detectar con métodos espaciales a causa de la naturaleza local de estas técnicas.

Para la segmentación basada en texturas se consideran tres características del espectro de Fourier que son útiles para la descripción de la textura; 1) picos prominentes del espectro que dan la dirección principal de los patrones de textura; 2) la localización de los picos en el plano de frecuencia da el período espacial fundamental de los patrones, y 3) la eliminación de los componentes periódicos mediante el filtrado deja elementos de la imagen no periódicos, que se pueden describir por técnicas estadísticas.

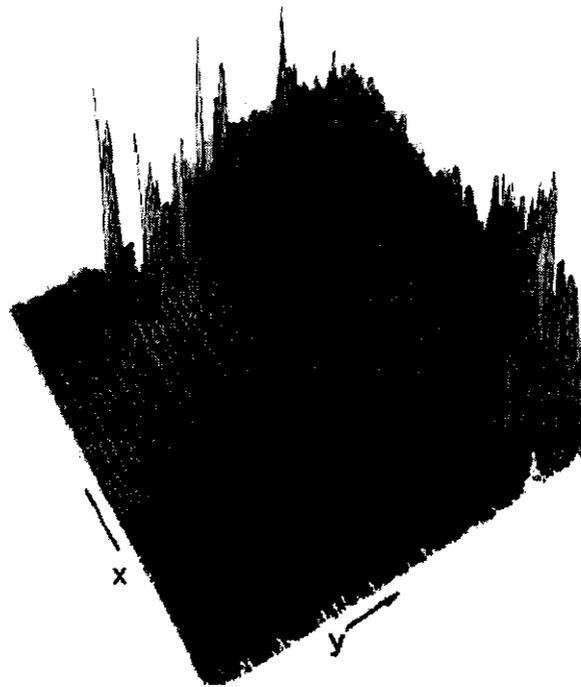
Dado que el espectro de una imagen real es simétrico con respecto al origen, así pues sólo se necesita considerar la mitad del plano de frecuencia. Por tanto, para efectos del análisis, cada patrón periódico está asociado con un solo pico del espectro, en lugar de dos. En la figura 2.5 se ven dos texturas así como los espectros de Fourier correspondientes, para ilustrar la forma de operar de este método.

2.7 DENSIDAD DE EXTREMOS RELATIVOS

Otro método que tiene su origen en los primeros años de la década de los setenta es el que sugiere que el número de extremos por unidad de área es una buena medida de la textura.



a)



b)

Figura 2.6. a) Imagen original. b) Extremos relativos de la imagen original. Puede notarse la diferencia en la densidad de extremos relativos entre las dos regiones de la imagen.

Normalmente, se trabaja en una imagen suavizada para eliminar los extremos, debido al ruido. El método más sencillo de esta familia sugiere definir los extremos en una sola dimensión a lo largo de un barrido horizontal.

Un problema que presenta contar simplemente todos los extremos en la misma meseta, es que los extremos por unidad de área no son sensibles a la diferencia entre una región que tiene pocas y grandes mesetas y una con muchos extremos de un píxel. La solución a este problema es contar solamente cada meseta una vez. Esto se logra localizando un píxel central en la meseta y marcándolo como el extremo asociado con la meseta. Otra manera de lograrlo es asociar el valor $1/N$ para cada valor en la meseta de extremos de N píxeles.

En el caso unidimensional se pueden asociar dos propiedades con cada extremo: su altura y su anchura. La altura de un máximo puede definirse como la diferencia entre el valor del máximo y el mínimo adyacente más alto. La profundidad de un mínimo puede definirse como la diferencia entre el valor del mínimo y el del máximo adyacente más bajo. La anchura de un máximo es la distancia entre dos mínimos adyacentes. La anchura de un mínimo es la distancia entre dos máximos adyacentes.

Los extremos bidimensionales son más complicados que los extremos unidimensionales. Una manera de encontrar los extremos en el sentido bidimensional amplio es el uso iterativo de operadores recursivos de vecindario que propagan valores extremos de modo adecuado. Las áreas máximamente conectadas de extremos relativos, pueden ser áreas de píxeles sencillos o mesetas de varios píxeles. Se puede marcar cada píxel de una región extrema relativa de tamaño N con un valor que indica que es parte de un extremo relativo de la altura correspondiente al valor con que se marcó o señalarlo con un valor que indica que es parte de una región extrema relativa de ese mismo valor. Alternativamente se puede marcar el píxel más centralmente localizado de la región extrema relativa con una cierta cantidad. A los píxeles no marcados puede

asignárseles el valor 0. Entonces para cualquier ventana específicamente centrada en un pixel dado, se les puede sumar los valores de todos los pixeles de la ventana. Esta suma dividida por el tamaño de la ventana es la altura promedio de los extremos del área. Alternativamente podíamos asignar una valor de 1 y la suma sería el número de extremos relativos por unidad de área asociada con ese pixel.

Más allá de sólo contar los extremos relativos, se pueden asociar propiedades con cada extremo relativo. Por ejemplo, dado un máximo relativo, se puede determinar el conjunto de pixeles alcanzable sólo por el máximo relativo dado, por ninguno otro, por caminos monotónicamente decrecientes. Este conjunto de pixeles alcanzables es una región conectada y forma una montaña. Los pixeles en la frontera pueden ser mínimos relativos o valles.

La altura relativa de una montaña es la diferencia entre su máximo relativo y el más alto de sus pixeles de frontera. El tamaño de una montaña es el número de pixeles que la componen. Su forma puede estar caracterizada por su circularidad y sus ejes de simetría. La circularidad se define como el cociente de la desviación estándar entre la media de las distancias del centro de la región a sus bordes. Los ejes de simetría pueden obtenerse erosionando la región hasta su esqueleto y contando el número de pixeles en el esqueleto. Para regiones elongadas, en el esqueleto puede verse la dirección de la elongación.

La figura 2.6 muestra una imagen y sus extremos relativos en las que esta serie de métodos es extremadamente útil.

2.8 AUTORREGRESION.

Hacia principios de los setenta, se propuso usar un modelo de este tipo, por primera vez. El modelo de autorregresión proporciona una forma de utilizar estimadores lineales del nivel de gris de un pixel, dados los niveles de gris en una

vecindad que lo contiene, para caracterizar la textura. Para texturas gruesas, todos los coeficientes serán similares. Para texturas finas los coeficientes variarán mucho. Debido a que una textura fina se caracteriza por su gran variabilidad en distancias relativamente cortas y una textura gruesa por su poco cambio en distancias cortas.

La dependencia lineal que un pixel de una imagen tiene en otro es bien conocida y puede ser ilustrada por la función de autocorrelación. Estos modelos utilizan el análisis de series de tiempo para calcular los parámetros de la textura.

Inicialmente lo que se propone es que dada una imagen sintética, el siguiente valor del nivel de gris para un pixel dado puede ser sintetizado como una combinación de los valores anteriores en la misma línea más una combinación de los valores aleatorios de ruido. Los coeficientes de las combinaciones lineales son los parámetros del modelo. Esta propuesta funciona muy bien teóricamente, sin embargo, en texturas reales, se logra un mejor desempeño con un modelo completamente bidimensional, que no solamente toma en cuenta los pixeles anteriores en la misma línea, sino además los pixeles que se encuentran arriba del pixel actual.

El poder de la solución del estimador de autorregresión lineal es el fácil uso del estimador en un modo que sintetiza las texturas partiendo de cualquier estimador lineal dado inicialmente. En este sentido, la solución autorregresiva es suficientemente capaz de capturar todos los detalles de una textura. Su punto débil es que las texturas que puede caracterizar son mayormente micro texturas.

Un trabajo interesante en esta línea de métodos es el de Gambotto (1981) quien propone usar series de tiempo bidimensionales para caracterizar la textura de una imagen. Usa muestras pequeñas y traslapa las ventanas. La primera de estas características permite distinguir las áreas homogéneas de la imagen, así

como las texturas que tienen características que varían lentamente. La segunda, se usa para encontrar las fronteras entre áreas homogéneas con mayor precisión.

2.9 MORFOLOGÍA MATEMÁTICA.

A mediados de los setenta se comenzó a usar la morfología para segmentar texturas. La morfología matemática es el estudio de la forma. Para el análisis de texturas, las formas analizadas son aquellas de las primitivas tonales. Una aproximación de este método requiere de la definición de un elemento estructurante y la generación de imágenes binarias que resultan de la traducción del elemento estructurante a través de la imagen y de la erosión de la imagen mediante el elemento estructurante. Los objetos de texturas pueden ser obtenidos de las nuevas imágenes binarias contando el número de píxeles con valor 1. Se ha encontrado un gran número de aplicaciones para este análisis cuantitativo de microestructuras, principalmente en ciencia de materiales y biología. Este proceso se ilustra en la figura 2.7

La importancia de esta solución para el análisis de texturas es que las propiedades obtenidas por la aplicación de los operadores en la morfología matemática pueden ser relacionados con las propiedades físicas de la forma tridimensional de los materiales cuya imagen se analiza.

El contexto de morfología matemática como una herramienta para extraer componentes de una imagen que sean útiles en la presentación y descripción de la forma de una región, tales como contornos, esqueletos y cerco convexo. También son de interés las técnicas morfológicas para el pre o postprocesado, tales como el filtrado morfológico, la reducción y el recortado.

El lenguaje de la morfología matemática es la teoría de conjuntos. Como tal, la morfología ofrece un método poderoso y único de abordar numerosos

problemas del procesado de imágenes. Los conjuntos en la morfología matemática representan las formas de objetos de una imagen.

2.10 CAMPOS ALEATORIOS MARCOVIANOS DISCRETOS.

También a mediados de los setenta se inició el uso de campos markovianos para diferenciar regiones texturizadas. El modelo de campo markoviano aleatorio para textura asume que el campo de textura es estocástico, estacionario y satisface una suposición de independencia condicional. Debido a que el campo es estacionario, la configuración de vecindad espacial es la misma sobre toda la imagen. Hay una dificultad obvia con que esta condición se cumpla en la frontera de la imagen. La forma usual de manejar el problema teóricamente es asumir que la imagen está pegada alrededor de un toroide. La suposición de independencia condicional es que la probabilidad condicional del pixel dados todos los pixeles restantes en la imagen es igual a la probabilidad condicional del pixel, dados solamente los pixeles en la vecindad.

En esta familia de modelos, se asignan dos variables aleatorias al pixel observado, una que caracteriza la intensidad y otra que etiqueta la textura de acuerdo a la ubicación correspondiente al pixel. Se usa el modelo del campo Markoviano aleatorio para la densidad condicional de la intensidad del campo dada la etiqueta del campo. Usando una distribución Markoviana discreta, se introduce información *a priori* sobre la etiqueta de la textura del campo. La segmentación puede formularse entonces como un problema de optimización que involucra la minimización de una función de energía de Gibbs. Es imposible una búsqueda exhaustiva por una solución óptima debido a la alta dimensionalidad del espacio de búsqueda. Por ejemplo, aún para el caso simple de segmentar una imagen de 128 x 128 en dos clases hay 2^{16384} configuraciones posibles de etiquetado. Por esta razón se utilizan por lo general algoritmos de relajación determinísticos.

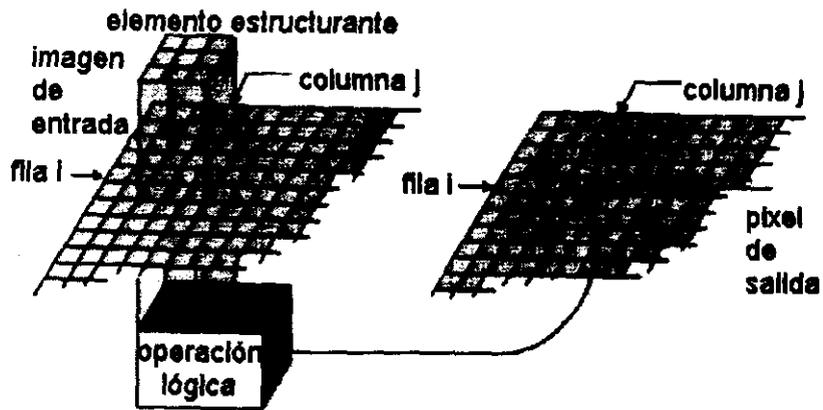


Figura 2.7 Forma de realizar un reconocimiento de texturas por morfología matemática.

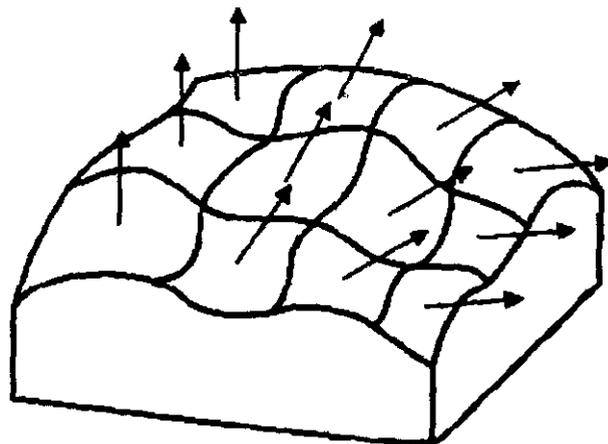


Figura 2.8 Esquematación de la forma de reconocimiento de texturas por dispersión de vectores.

Algunas de las modificaciones que se han propuesto a este método incluyen la de Geman y Geman (1984) quienes proponen una relajación estocástica relacionada con el método de Monte Carlo, concretamente una versión del algoritmo *Metrópolis*. Posteriormente, Geman, Geman y Graffigne (1988) construyen un campo markoviano aleatorio con una distribución previa de las intensidades (observadas) y las etiquetas (no observadas). En su propuesta las etiquetas pueden asignarse ya sea a texturas o a los indicadores de fronteras.

Otro método relacionado es el de Marroquín (1989, 1991) quién sugiere que la construcción del campo aleatorio markoviano se haga en un red finita cuyas configuraciones de equilibrio consisten de líneas rectas divididas en trozos cuyas orientaciones son arbitrarias. Utiliza sólo las interacciones con los vecinos más cercanos.

La propuesta de Chang y Chatterjee (1992) utiliza una solución híbrida que combina técnicas de clasificación estadísticas y estructurales. Se usa un modelado explícito a través de campos aleatorios de Gibbs y una algoritmo de relajación semideterminístico.

Recientemente Krishnamachari y Chellapa (1997) presentan un modelo de segmentación de texturas usando un campo de Gauss Markov con multiresolución.

2.11 FUNCION DE AUTOCORRELACIÓN

Hacia fines de los setenta se comenzó a usar la función de autocorrelación para dividir una imagen, basado en la textura. Se considera que la textura se relaciona con el tamaño espacial de los niveles de gris primitivos de una imagen. Los niveles de gris primitivos de mayor tamaño indican texturas más gruesas; los niveles de gris primitivos de menos tamaño pertenecen a texturas más finas. La

función de autocorrelación es una característica que describe el tamaño de los niveles de gris primitivos.

La fortaleza de esta aproximación a la textura es la familiaridad que se tiene con estos conceptos. Sin embargo, uno de los problemas inherentes es la calibración de la imagen respecto al nivel de gris. Los procedimientos son invariables aún en una transformación monotónica de nivel de gris. Para compensar esto, puede usarse la cuantización de la probabilidad. Pero el precio pagado por la invariabilidad de las imágenes cuantizadas bajo transformaciones de niveles de gris da como resultado la pérdida de precisión de nivel de gris en la imagen cuantizada.

Desde un punto de vista, la textura se relaciona con el tamaño espacial de las primitivas tonales de la imagen. Las primitivas tonales de mayor tamaño son indicadoras de texturas más gruesas; las primitivas tonales de menor tamaño son indicadoras de textura más fina. Si las primitivas de niveles de gris son espacialmente periódicas, la función de autocorrelación caerá y se levantará en forma periódica. La función de autocorrelación de un objeto que dice algo sobre el tamaño de las primitivas tonales.

La primitiva del nivel de gris en la autocorrelación es el nivel de gris. La organización espacial se caracteriza por el coeficiente de correlación que es una medida de la dependencia lineal que un pixel tiene con otro.

2.12 DISPERSION DE VECTORES.

También hacia fines de los 70 surgieron los primeros intentos de usar la dispersión de los vectores para reconocer las texturas de una imagen. En la técnica de dispersión de vectores, la textura de la imagen se divide en vecindarios mutuamente exclusivos y un plano inclinado adecuado a los niveles de gris se calcula para cada vecindario, como se ve en la figura 2.8. Es decir, se usa un

sistema coordinado relativo cuyo origen es el centro del vecindario que permite modelar los niveles de gris del vecindario por medio de una ecuación. Lo que se hace entonces es calcular el vector normal al plano y la serie de normales a los planos es lo que propiamente describe la textura.

2.13 ENERGIA DE LAS TEXTURAS.

A principios de los 80 surge el primer intento de usar la energía de las texturas, como el último de los métodos que se detallan aquí.

Esta es la aproximación de transformación de la energía de la textura, propuesta por Laws (1980). Se aplica una variedad de operadores lineales a una imagen. Cada operador tiene una pequeña vecindad para su dominio. Las salidas del operador cuadrado son promediadas con una ventana de pesos iguales que tiene un dominio espacial mayor que los operadores originales. Otros autores notaron que es posible utilizar las transformadas ortogonales discretas como seno o coseno discretos aplicadas localmente a cada vecindad de pixel en lugar de usar operadores *ad hoc*.

En la solución para la energía de texturas Laws la imagen es primero convolucionada con una variedad de kernels que se discuten en el siguiente capítulo. Entonces cada imagen convolucionada se procesa con un operador no lineal para determinar la energía total de textura en la vecindad de cada pixel 7×7 . La solución de la energía de la textura se parece mucho a la de la transformación, pero usa ventanas más pequeñas o apoyo de los vecinos. Los kernels que Laws usa tienen soportes para vecindarios de 3×3 , 5×5 , 7×7 .

La mayor dificultad para la solución de energía de las texturas es la posibilidad de introducir errores significativos a lo largo de las fronteras entre diferentes texturas, porque es exactamente para las posiciones de las fronteras que el soporte de los vecinos incluye una mezcla de texturas. Es posible que el

vector de energía de las texturas de un vecindario de tales texturas mixtas esté cerca de un vector prototipo de una tercera textura que no tiene nada que ver con las texturas de la mezcla.

Para resolver este problema, Hsiao y Sawchuck (1989a, 1989b) llevan a cabo más de un nivel de procesamiento para cada imagen de energía de textura. Para cada posición de pixel de la imagen calculan la media y la varianza de cuatro vecindarios de 15 x 15 que son respectivamente el sureste, el suroeste, el noreste y el noroeste. Luego crean una imagen de energía de las texturas suavizada.

Unser y Eden (1989, 1990) tratan la modificación multiescalar de la solución de energía de textura. Ellos toman la imagen de energía de textura iterativamente la suavizan con kernels gaussianos con una progresión en la escala de media octava. Reducen la característica de dimensionalidad diagonalizando simultáneamente las matrices dispersas y dos niveles sucesivos de resolución espacial.

Este método es discutido con mayor detalle en el siguiente capítulo. Se incluyen ahí las máscaras de convolución de Laws, así como una descripción más amplia de los pasos que componen a este método.

2.14 SEGMENTACION BASADA EN TEXTURAS.

La mayor parte del trabajo con las características de textura ha sido dirigido al análisis de textura de una imagen. Esto se refleja, por ejemplo, en la gran cantidad de métodos que han surgido hasta el momento y en que de hecho es un área en la que se sigue trabajando todavía porque no hay un método perfecto, sino diferentes soluciones *ad hoc*.

Sin embargo, es claro que una imagen no necesariamente tiene una textura homogénea. Esta es la razón por la cual una operación importante de

procesamiento de imágenes es la segmentación de imágenes en regiones, cada una de las cuales debe tener una textura homogénea.

En la figura 2.9 se muestra un ejemplo de las regiones que pueden componer una imagen. La restricción es clara, cada par de regiones adyacentes debe tener diferente textura y los bordes deben ser lo menos rasgados posible.

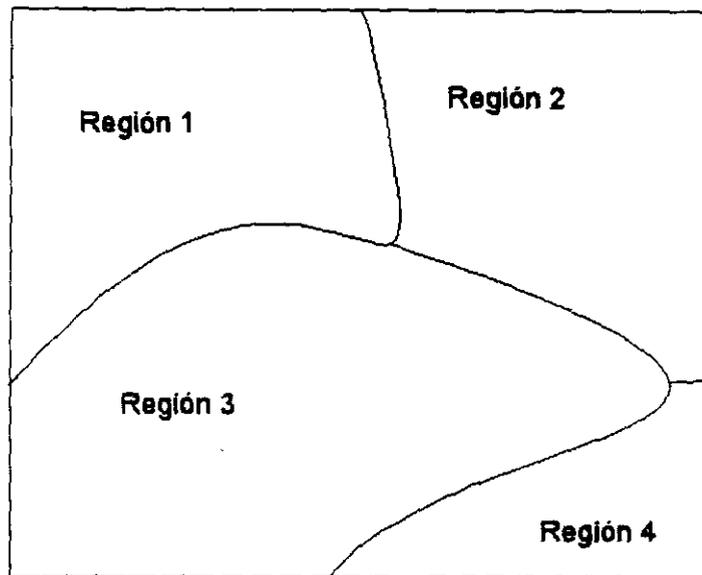


Figura 2.9 Diferentes regiones con textura que pueden componer a una imagen.

Los descriptores de cada región incluyen un indicador de si la textura es isotrópica, que significa que tiene las mismas propiedades en todas las direcciones o direccional, cuyo significado es que predomina una de las direcciones; el tamaño del elemento de textura, y la separación entre elementos de textura. Si la textura fue considerada direccional, entonces la textura incluye la orientación.

Con base en esos descriptores y a las técnicas de segmentación descritas en el capítulo anterior es que puede dividirse la imagen en las regiones que la componen.

La segmentación de imágenes es una etapa preliminar esencial en la mayor parte de los problemas de reconocimiento de formas y análisis de escenas. La elección de una técnica de segmentación en lugar de otra, y de un modelo de análisis de texturas en lugar de otro, está dictado principalmente por las características del problemas que se está considerando

CAPITULO III: DESCRIPCION DE UN METODO PARA SEGMENTACION BASADA EN TEXTURAS.

Lo que se pretende lograr al proponer un nuevo método para la segmentación basada en la textura es:

1. Mejorar los resultados de la segmentación de imágenes con textura, especialmente a lo largo de las fronteras.
2. Mejorar la segmentación de los interiores de las regiones, tomando en cuenta las relaciones espaciales entre los píxeles.

Se propone un método que aprovecha la energía de las texturas en la extracción de objetos. El método se basa en un concepto de suavizado adaptativo del ruido que toma en cuenta la naturaleza no estacionaria del problema (Laws (1980) y Hsiao y Sawchuk (1989a y 1989b)) y en la relajación probabilística basada en el método de vertientes (Hansen y Higgins (1997)).

Las etapas que componen este método son las siguientes:

1. La energía de las texturas se estima primero usando una ventana de tamaño pequeño para reducir la posibilidad de mezclar las estadísticas a lo largo de los bordes de la región.
2. Los valores estimados de la energía de las texturas son suavizados entonces por un método de filtrado para reducir la variabilidad de las estimaciones mientras se retiene la exactitud de los bordes.
3. Los valores estimados son usados en un algoritmo de agrupamiento k-means para estimar las estadísticas de las clases.
4. Las estadísticas estimadas de las clases son usadas por un clasificador bayesiano para realizar un etiquetado probabilístico inicial.
5. Las restricciones espaciales se refuerzan a través del uso de un algoritmo de relajación probabilística. Para que las iteraciones de la relajación probabilística sean más eficientes se usa el método de vertientes.

El resultado que se obtiene es un mapa que indica a que clase pertenece cada uno de los pixeles que componen la imagen.

3.1 ENERGIA DE LAS TEXTURAS

Esta es la aproximación de transformación de la energía de la textura, propuesta por Laws (1980). Se aplica una variedad de operadores lineales a una imagen. Cada operador tiene una pequeña vecindad para su dominio. Las salidas del operador cuadrado son promediadas con una ventana de pesos iguales que tiene un dominio espacial mayor que los operadores originales.

En la solución para la energía de texturas Laws la imagen es primero convolucionada con una variedad de operadores lineales llamados kernels que se mencionan más adelante. Entonces cada imagen convolucionada se procesa con un operador no lineal para determinar la energía total de textura en la vecindad de cada pixel 7×7 . El método de la energía de la textura se parece mucho la de la transformación, pero usa ventanas más pequeñas o apoyo de los vecinos. Los kernels que Laws usa tienen soportes para vecindarios de 3×3 , 5×5 , 7×7 .

La mayor dificultad para el método que utiliza la energía de las texturas es la posibilidad de introducir errores significativos a lo largo de las fronteras entre diferentes texturas, porque es exactamente para las posiciones de las fronteras que el soporte de los vecinos incluye una mezcla de texturas. Es posible que el vector de energía de las texturas de un vecindario de texturas mixtas esté cerca de un vector prototipo de una tercera textura que no tiene nada que ver con las texturas de la mezcla.

Como se mencionó en el capítulo anterior, la propuesta de Laws (1980) para la caracterización de texturas consiste de dos pasos. Primero, un conjunto de microtexturas denotadas por $g(i,j)$ se calculan usando máscaras de convolución de suma cero 3×3 , 5×5 o 7×7 . Estas máscaras están diseñadas para actuar como

filtros para ciertos tipos de variaciones cuasiperiódicas que se encuentran comúnmente en las imágenes con textura. El segundo paso es obtener estadísticas en ventanas grandes. En este trabajo se encontró que las estadísticas más útiles son varianzas locales de la muestra. A las características de la varianza obtenidas por medio de diferentes máscaras de convolución, Laws (1980) les llamó medidas de la energía de las texturas.

Una vez calculada la energía de la textura, Laws (1980) usó una característica de selección opcional para reducir la dimensionalidad del proceso de clasificación que consiste en no tomar todos los píxeles, sino sistemáticamente algunas filas y columnas de la imagen. Hay dos clases de procedimientos que se usan, uno de ellos si se dispone de los prototipos etiquetados para las clases y sus estadísticas y otro en el caso de que no se conozcan. Si se conocen las etiquetas de los prototipos, el método de análisis de discriminante múltiple se usa para reducir el espacio de las características. Luego, un clasificador basado en la mínima distancia a los centroides se usa para asignar la etiqueta de observación de prueba. En este procedimiento no se incluye información espacial.

Para calcular la energía de las texturas, Laws (1980) usó kernels, cuyo nombre lleva las letras L, E, S, W, R y O seguido de un número que indica la dimensionalidad del vector. En el estudio de Laws (1980), L se refiere a nivel (level), E quiere decir borde (edge), S indica forma (shape), W sirve para onda (wave), R implica ondulación (ripple) y O es la abreviatura de oscilación (oscillation).

Las formas bidimensionales son generadas de las formas unidimensionales por producto vectorial (los kernels unidimensionales aparecen en la figura 3.1). Esto es, si k_1 y k_2 son dos formas unidimensionales, cada uno de ellos es un vector renglón con K columnas, entonces $k_1 \cdot k_2$ constituye un kernel $K \times K$. Laws (1980) encontró especialmente efectivos los kernels que se muestran en la figura

3.2

$$\begin{aligned} L3 &= [1 & 2 & 1 &] \\ E3 &= [-1 & 0 & 1 &] \\ S3 &= [-1 & 2 & -1 &] \end{aligned}$$

a)

$$\begin{aligned} L5 &= [1 & 4 & 6 & 4 & 1 &] \\ E5 &= [-1 & -2 & 0 & 2 & 1 &] \\ S5 &= [-1 & 0 & 2 & 0 & -1 &] \\ W5 &= [-1 & 2 & 0 & -2 & 1 &] \\ R5 &= [1 & -4 & 6 & -4 & 1 &] \end{aligned}$$

b)

$$\begin{aligned} L7 &= [1 & 6 & 15 & 20 & 15 & 6 & 1 &] \\ E7 &= [1 & -4 & -5 & 0 & 5 & 4 & 1 &] \\ S7 &= [-1 & -2 & 1 & 4 & 1 & -2 & -1 &] \\ W7 &= [-1 & 0 & 3 & 0 & -3 & 0 & 1 &] \\ R7 &= [1 & -2 & -1 & 4 & -1 & -2 & 1 &] \\ O7 &= [-1 & 6 & -15 & 20 & -15 & 6 & -1 &] \end{aligned}$$

c)

Figura 3.1 Kernels unidimensionales de la energía de texturas propuestos por Laws (1980). a) kernels para 3 x 3; b) kernels para 5 x 5 c) kernels para 7 x 7.

$$E5L5 = \begin{pmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

$$E5S5 = \begin{pmatrix} -1 & 0 & 2 & 0 & -1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & -4 & 0 & 2 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

$$R5R5 = \begin{pmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}$$

$$L5S5 = \begin{pmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{pmatrix}$$

Figura 3.2 Cuatro de las máscaras de microtextura propuestas por Laws (1980). E5L5 resalta los bordes y el nivel de gris. R5R5 subraya las ondulaciones. E5S5 se encarga de remarcar los bordes y las formas y L5S5 enfatiza los niveles de gris y la forma.

3.2 USO DE SUAVIZADO DE CARACTERÍSTICAS Y RELAJACION PROBABILÍSTICA.

Hsiao y Sawchuk (1989a, 1989b) mencionan que una de las razones principales de errores de segmentación a lo largo de los bordes de la región es el método de formación de la energía de la textura en un promedio de una ventana móvil de los valores absolutos de las características de microtexturas, aunque las operaciones de dichas ventanas móviles son simples y rápidas, introducen errores significativos a lo largo de los bordes. Su razón es la superposición del promedio de ventana en los bordes de textura. Cuando ocurre esto, las estadísticas resultantes son la mezcla de dos juegos de estadísticas que pueden recordar otra clase de textura muy diferente. En este caso, los píxeles fronterizos son incorrectamente clasificados como una tercera clase, diferente de las dos ventanas interiores sobrepuestas.

Una manera de evitar este problema, se encuentra en el método propuesto por Hsiao y Sawchuk (1989a, 1989b). Esta solución consiste en calcular más cuidadosamente la desviación estándar de la muestra de las imágenes filtradas de microtextura. Se puede percibir el proceso de obtener las características estadísticas como el problema de suavizar una imagen ruidosa. Al suavizar el ruido siempre se encuentra el siguiente problema: cómo suavizar el ruido sin desenfocar los bordes. Hay muchas soluciones al problema de suavizar el ruido; ahora el enfoque será a algunos de los conceptos para la extracción de características.

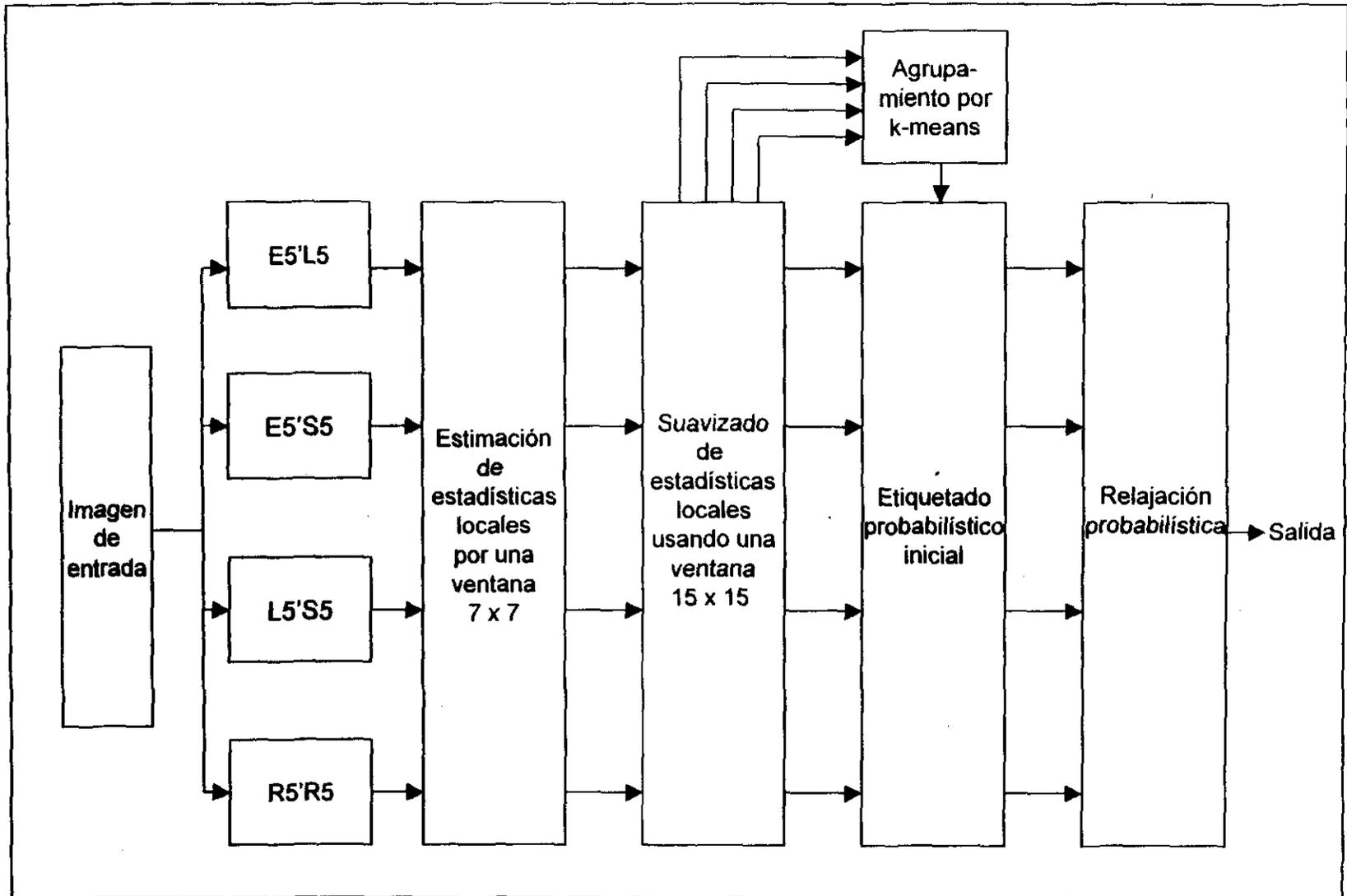


Figura 3.3 Diagrama de bloques del sistema de segmentación basado en texturas.
Tomada de Hsiao y Sawchuk (1989b)

Aquellas ventanas que contienen regiones fronterizas, generalmente tienen una variabilidad más alta causada por los bordes. Así, se define la estadística final de textura como la media de la muestra que corresponde a la variación de la muestra que es la más baja entre cuatro cuadrantes. Otra cuestión que necesita ser considerada es el efecto de usar diferentes tamaños de ventanas para calcular las estadísticas de muestras locales. Esto implica aceptar que la clase de texturas con las que se trabaja es de grano fino.

La opción de tamaño de ventanas usadas para calcular las estadísticas locales es un problema de "negociación". Usando tamaños pequeños (7×7) es posible calcular las estadísticas con mayor precisión cerca de los bordes y en las regiones pequeñas, pero con tan pocos datos, solamente cuarenta y nueve, las estadísticas son más pobres que si se usa una ventana grande debido al menor número de puntos. Por otro lado, usando ventanas grandes (15×15) se tienen más puntos de datos para reducir la variabilidad estadística de los cálculos en las regiones grandes, pero los resultados cerca de los bordes entre las texturas y en las regiones pequeñas son escasos.

El diagrama de bloques que aparece en la figura 3.3 corresponde al proceso de segmentación no supervisada por el método propuesto por Hsiao y Sawchuk (1989 a y b). Cabe resaltar que este método supone que las características de energía de textura en cada pixel han sido calculadas y que se conoce el número de clases. Con la finalidad de reducir la cantidad de cómputo necesaria para resolver el problema, el algoritmo de agrupamiento trabaja sobre un subgrupo de datos que consiste de cada cuarta línea y cada cuarto pixel de los planos característicos.

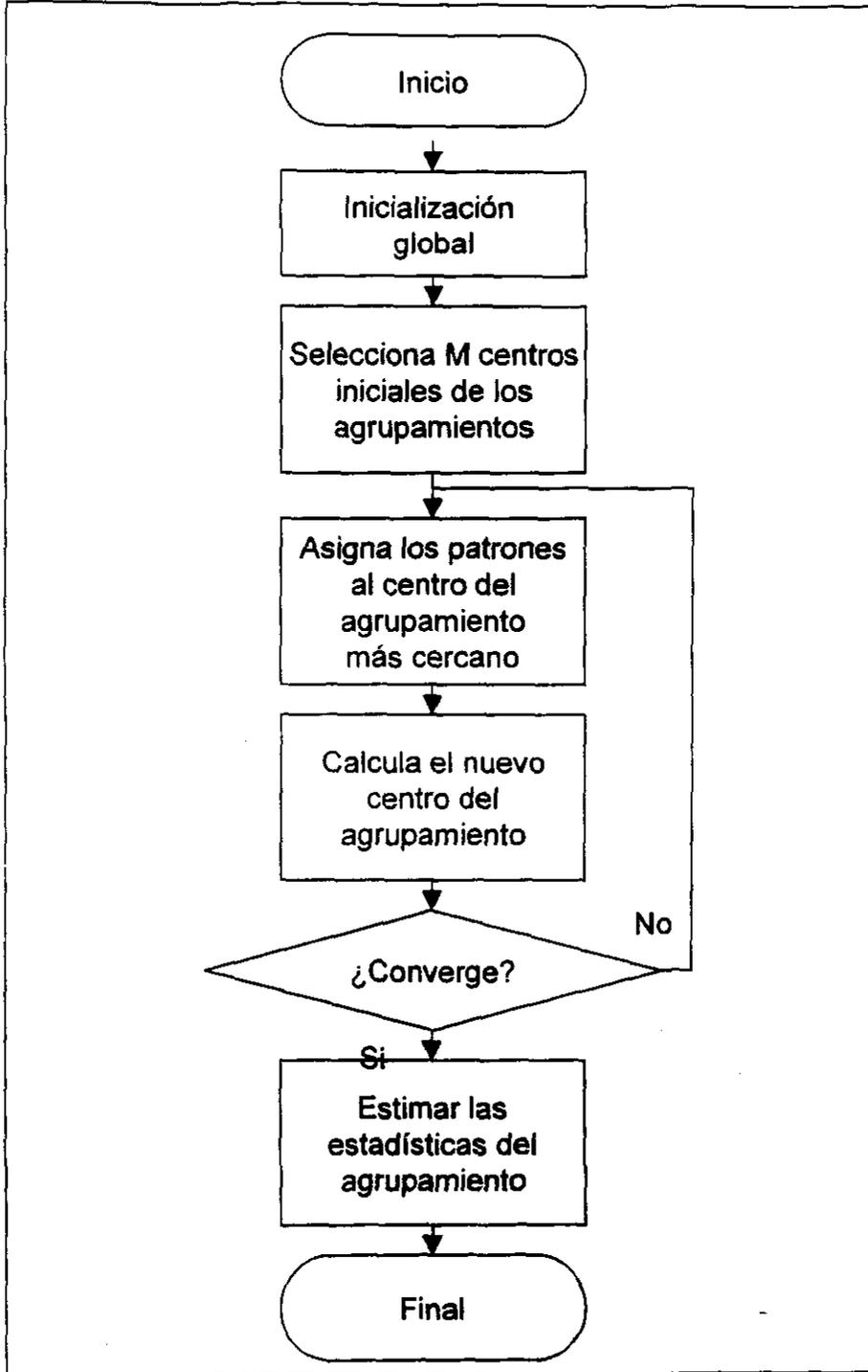


Figura 3.4 Diagrama de flujo del proceso de agrupamiento. Tomada de Hsiao y Sawchuk (1989b).

Para llevar a cabo el agrupamiento, el primer punto inicial es el vector medio total del conjunto de datos, y los puntos subsiguientes son seleccionados examinando los patrones de su secuencia de entrada. Solamente el patrón que se haya cuando menos a alguna distancia especificada de todos los puntos elegidos es aceptado. En la Figura 3.4 se muestra el diagrama de flujo de la solución de agrupamiento total.

El nuevo centro del agrupamiento es sencillamente el promedio de todos los patrones asignados a tal agrupamiento. La iteración continúa hasta que el proceso converja o el número de iteraciones exceda un límite proporcionado por el usuario, que en el estudio de Hsiao y Sawchuk (1989a, 1989b) fue de 25 iteraciones. Se acepta que la convergencia ha sido alcanzada cuando la diferencia de las medias de todos los agrupamientos de una iteración a la siguiente es menor que una cota de error.

El método de relajación probabilística fue usado como medio para reforzar las restricciones espaciales en el algoritmo de segmentación. El proceso de relajación probabilística está relacionado con un grupo de objetos y un grupo de etiquetas de clase.

El algoritmo de relajación probabilística usado se compone de los siguientes pasos:

1. Para cada uno de los ocho vecinos conectados se usan ecuaciones para calcular los coeficientes de compatibilidad. Estos coeficientes se fijan durante el proceso iterativo.
2. Sigue el proceso iterativo de actualización usando ecuaciones en que se haga un muestreo el efecto de pares de uno de los ocho vecinos conectados.
3. Ya que cada pixel tiene ocho vecinos, hay un promedio de estas ocho lecturas.
4. Repetir los pasos 2 y 3 hasta encontrar el criterio de terminación.

Hay varias guías para evaluar el funcionamiento del algoritmo de relajación. La suma de las diferencias absolutas de probabilidad de pertenecer a una clase debe ser más pequeña a partir de algunas iteraciones. Las probabilidades finales no deberían ser demasiado distantes en promedio de las iniciales; no sería satisfactorio el proceso si nos llevara a un juego arbitrario de probabilidades finales sin relación con las iniciales. La entropía de la clasificación probabilística, después de aplicar la relajación, debería ser menor que la entropía inicial.

Algunos autores sugieren que el vector de probabilidades debe estar asociado con cada pixel. Estas probabilidades definen un rango variable que representan las posibles etiquetas de pixel. Las probabilidades de pixeles vecinos se usan iterativamente para actualizar las probabilidades de un pixel dado basado en las relaciones estadísticas entre etiquetas de pixel.

El algoritmo usado para estudiar los efectos de limitar el espacio de decisión de división de probabilidad es el siguiente:

1. Elegir el valor de la división de probabilidades.
2. Usar el algoritmo de agrupamiento k-means para formar agrupamientos del espacio característico. Una vez formados los agrupamientos, pueden calcularse los datos estadísticos tales como vectores medios y matrices de covarianza. Se usaron estas estadísticas de agrupamiento no sesgadas para el propósito de etiquetado probabilístico inicial.
3. Basado en las estadísticas iniciales de agrupamiento y aceptando una distribución normal, las probabilidades de etiquetado de agrupamiento son asignadas a cada pixel.
4. Cuando las probabilidades de etiquetado del agrupamiento son menores que el valor de segmentación de probabilidad este se pone en cero y luego se normalizan las probabilidades diferentes de cero.
5. A partir del etiquetado de probabilidad inicial de partición, puede generarse una tabla de compatibilidad.

6. Se genera una máscara para cada clase que indica aquellos píxeles con probabilidad cero o uno. Esta máscara puede usarse para guiar el proceso de relajación y trabajar sobre esos píxeles cuyas probabilidades no son cero o uno. Además, la cantidad de datos que necesita ser procesada también puede calcularse a partir de esta máscara.

7. Con el proceso inicial de etiquetado de probabilidad y los coeficientes de compatibilidad a la vista, se puede comenzar el proceso de relajación probabilística.

8. Después de cada iteración o de unas cuantas iteraciones, puede calcularse el coeficiente de clarificación. Debe repetirse el proceso de relajación hasta llegar al criterio prefijado de terminación.

3.3 METODO DE VERTIENTES

Al reconocer objetos en una imagen binaria, todo lo que hace una computadora es interpretar como parte de un objeto a todos los píxeles encendidos que están conectados entre sí. Aunque para el ojo humano pudiera ser claro que se trata de dos objetos unidos, si el sistema encuentra que los píxeles que los componen están conectados, los tratará como a un sólo objeto y para uno sólo calculará parámetros por su posición y su forma, lo que obviamente será incorrecto. Por esto, es deseable reconocer estos casos de objetos unidos y separarlos para procesarlos adecuadamente.

Existe un método para lograr esto, conocido como segmentación convexa o segmentación por vertientes. Es un método general que produce la separación deseada bajo las condiciones siguientes:

1.- Los objetos deben ser localmente convexos, de tal forma que necesiten ser separados.

2.- El grado de unión o sobreposición de los objetos debe ser suficientemente pequeño para que pueda encontrarse la línea de separación.

La implantación de este método que reportó Santiago et al (1992) está basado en el método de segmentación por vertientes que fue propuesto por Beucher y Lantéjoul (1979) y por Lantéjoul y Beucher (1981). La propuesta es crear un mapa de distancias. Se usa una "imagen auxiliar" en la cual el brillo del pixel es una medida de cuán lejos de la periferia del objeto está el pixel binario correspondiente a la imagen original. En la práctica, generalmente se usa la simplificación de medir la distancia en cuatro direcciones ortogonales y cuatro diagonales o cuatro ortogonales, y seleccionar la menor de ellas como la distancia. Su valor se almacena como el brillo del pixel correspondiente.

Si no se proporciona una imagen binaria, es necesario indicar el umbral de discriminación, es decir, determinar a partir de qué valor del brillo del pixel se considera que es parte de un objeto y por tanto, todo lo demás será considerado fondo. De esta forma, el sistema trabajará como si la imagen fuera binaria.

El método para crear el mapa de distancias, que fue implantado y se detalla en el artículo de Santiago et al (1992), es el propuesto por Russ (1990) y es muy eficiente porque requiere solamente rastrear la imagen cuatro veces. Idealmente la imagen debe ser binaria. En caso contrario se toma un valor de corte. En el caso de los píxeles que tienen un valor por encima del valor de corte en la imagen binaria, se les asigna el valor máximo y se asigna cero a todos los demás. Empezando de la esquina superior izquierda y recorriendo la imagen línea por línea de arriba hacia abajo y de izquierda a derecha, la regla a seguir es que ningún punto puede ser más de una unidad mayor que el menor de sus vecinos. Esto se repite línea por línea de abajo hacia arriba y de derecha a izquierda. No importando el tamaño del objeto, su forma o complejidad, dos repeticiones completas de este procedimiento, en ambas direcciones, sobre la imagen, producen el mapa de distancias. Esto requiere un tiempo fijo. Se puede ver esto en la figura 3.5.

Para obtener una mejor aproximación de los resultados a círculos isotrópicos, en esta implantación se restringe la diferencia de brillo para vecinos horizontales y verticales a 2 y para vecinos diagonales a 3. Al terminar de calcular el mapa de alturas, los valores se dividen entre dos, redondeando hacia abajo, para ayudar a compensar la pequeña diferencia entre $3/2$ u $\sqrt{2}$.

Los objetos superpuestos son segmentados a lo largo de las líneas de las vertientes en el mapa de distancias. Esto se entiende mejor imaginando que el brillo de cada pixel es una altura física, de tal forma que los puntos brillantes en el centro de los objetos son montañas, esto se ilustra en la figura 3.6. Las vertientes son los valles entre montañas. Cuando los picos son de la misma altura (significando que los objetos son del mismo tamaño), el punto de división está a la mitad del camino entre los dos picos, pero cuando un objeto es más pequeño que el otro, el pico es menor y la vertiente cae más cerca del pico menor. El método asume que los objetos son convexos, de tal manera que deben ser segmentados, y también asume que el grado de unión o sobreposición es suficientemente pequeño para que exista un valle entre los picos en el centro de cada objeto en el mapa de distancia codificado por brillo.

Para encontrar las líneas de la vertiente, en el mapa de distancias, las mesetas forman el núcleo de crecimiento subsecuente por engrosamiento; sin embargo, caen a diferentes niveles (representando el centro de objetos de diferentes tamaños). Se toma en cuenta esto en la rutina de engrosamiento descrita abajo, de tal manera que es posible poner inmediatamente todas las mesetas en la imagen binaria final.

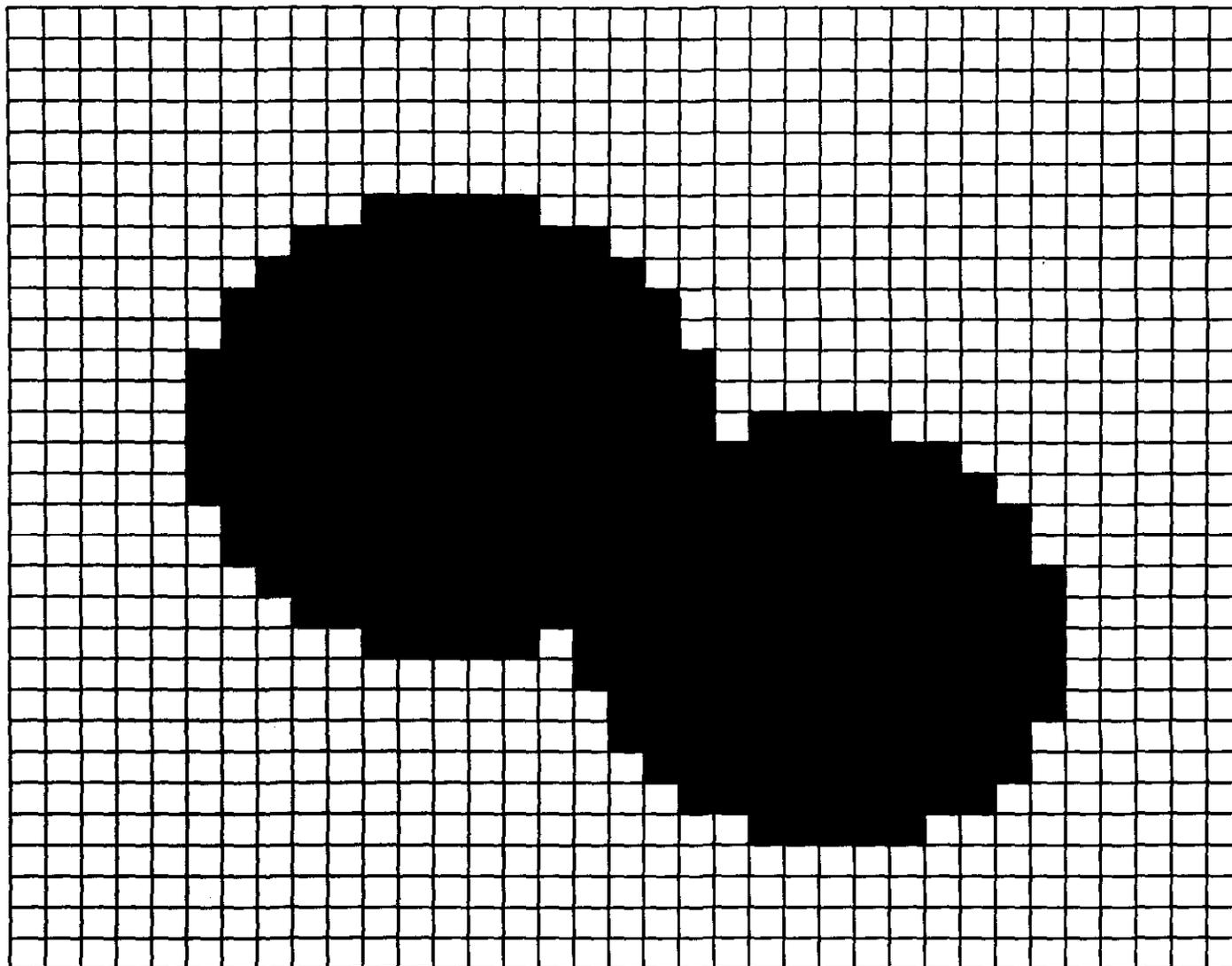


Figura 3.5 Ejemplo de mapa de distancias.

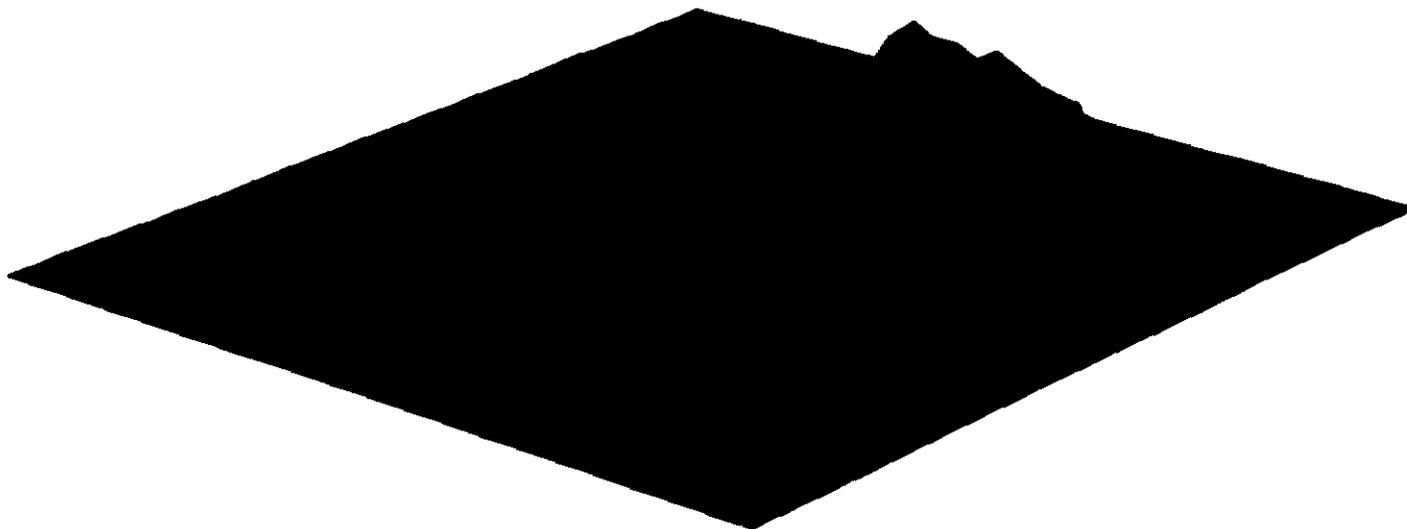


Figura 3.6 El brillo de cada pixel representa una altura física. La separación entre las dos "montañas"

El engrosamiento se lleva a cabo en pasos, empezando a la altura uno menos que la meseta más alta. En cada altura, los píxeles que tengan al menos ese valor son candidatos para engrosamiento. Se volverán parte de la imagen, es decir se igualarán a la meseta más alta, si no crean una nueva unión entre cualquier par de píxeles vecinos. Esta operación requiere de la aplicación de reglas de lógica. Una tabla de 256 posibles configuraciones de los 8 vecinos alrededor de cada píxel se almacena en memoria. En cada posición de la tabla hay 8 direcciones de crecimiento del píxel considerado, y por cada una de ellas un bit decide el destino del píxel. Los patrones de vecinos que serían conectados si se encendiera el bit central no se permiten, todos los demás sí.

La tabla de destino es aplicada a cada píxel en la imagen cuyo brillo está en el valor especificado de altura o distancia, y se repite secuencialmente para la imagen en cada una de las ocho posibles direcciones de crecimiento. En la práctica, la mayoría de los patrones de píxeles se llenan de acuerdo a las reglas para algunas direcciones, y por tanto la mayoría de los píxeles de cada nivel de altura se llenan en los primeros pocos pasos. Es necesario que el proceso se repita hasta que no sea posible hacer más cambios.

La operación anteriormente descrita se repite en cada nivel de brillo inferior, hasta terminar. La imagen binaria resultante es recuperada y puede ser comparada con la original, de la que difiere sólo en la remoción de píxeles a lo largo de las líneas para separar objetos unidos. Las líneas de los píxeles quitados equivalen a las uniones de conos en el mapa de altura, cada cono construido alrededor de una meseta o conjunto últimamente erosionado, y creciendo de una base en la imagen binaria original. La segmentación tiene lugar a lo largo de las líneas de vertiente en las uniones.

La creación del mapa de distancias resulta ser un punto crítico, dado que, sobre la base de él se hace la separación de los objetos. Un mapa

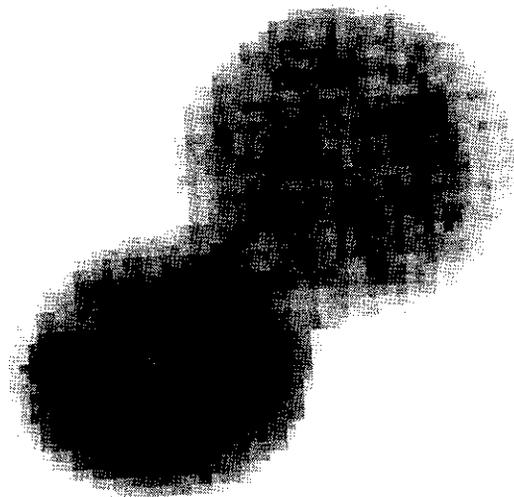
incorrectamente construido puede provocar que se separen objetos que no deberían o que no se separen los que sí deberían hacerlo.

Para crear el mapa de distancias, se puede usar la aplicación secuencial de operadores morfológicos. Los objetos en la imagen se erosionan secuencialmente, y después de cada iteración los valores de brillo de todos los píxeles restantes son incrementados en uno. Al completar el proceso (cuando la imagen ha sido completamente erosionada), se habrá construido el mapa de distancias correspondiente.

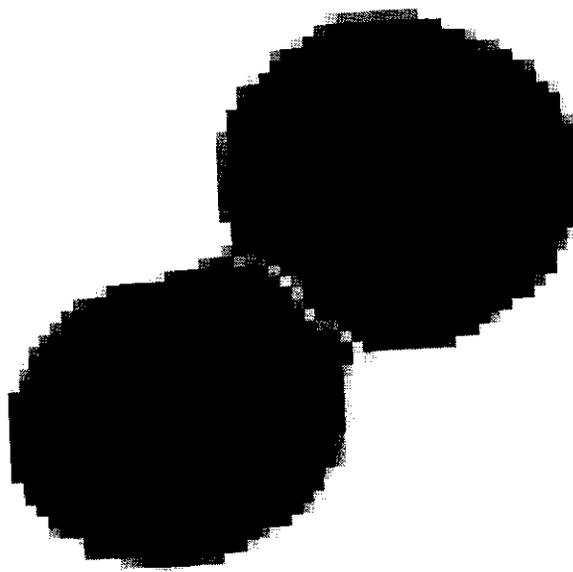
En cada paso, cúmulos aislados de puntos que desaparecerán completamente en el siguiente paso se identifican y almacenan. A partir del último estado, estos conjuntos erosionados últimamente son engrosados. El proceso de engrosamiento es similar al de dilatación, pero se modifica para eliminar puntos que pudieran causar la formación de nuevas conexiones.

El engrosado es iterativo, el píxel central es rellenado solamente si no se conectarán dos píxeles vecinos. La clave de este método es encontrar todos los puntos erosionados al final, que son las islas finales, quitadas últimamente por erosión iterativa. Es importante entender que estas no aparecen al mismo tiempo en la secuencia de erosión, por que en general los objetos no son del mismo tamaño.

Esta serie de pasos, aparentemente simple, se enfrenta a algunos problemas prácticos importantes. El primero y principal, es que el proceso es iterativo. Cada uno de los pasos de erosión y engrosado debe repetirse hasta que ya no haya más cambios en la imagen. Existen algunas formas que pueden tomar muchas repeticiones. En segundo lugar, como el método requiere que la imagen binaria entera en cada paso del proceso de erosión esté disponible en el proceso de engrosado, ya sea con almacenamiento masivo o muchas repeticiones del proceso de erosión.



a)



b)

Figura 3.7 a) Imagen original. b) Imagen segmentada, por el método de vertientes, de dos células pegadas.

El método implantado funciona adecuadamente, es más rápido y eficiente que el que usa erosión y dilatación secuencialmente; sin embargo, si durante el proceso de engrosamiento se usaran banderas para saber en qué líneas de la imagen todavía hay píxeles por revisar con valor igual al de la meseta considerada, según sugiere Russ (1990) se aceleraría aún más el proceso.

Una de las desventajas de este método es que sólo puede aplicarse si los objetos son localmente convexos y el grado de unión o sobreposición de los mismos es suficientemente pequeño para que pueda encontrarse la línea de separación. Es importante comentar que las figuras toroidales no son segmentadas adecuadamente, debido a la forma en que se construye el mapa de distancias, se suelen dividir en varios fragmentos. Una imagen original y los resultados de aplicar este método a dicha imagen se muestran en la figura 3.7.

3.4 METODOS DE RELAJACION PARA SEGMENTACION SUPERVISADA DE IMAGENES

Hanssen y Higgins (1997) no trabajan en segmentación basada en texturas, sino solamente en segmentación de imágenes. Ellos encuentran que el etiquetado de relajación se ha aplicado en muchos usos de visión por computadora tales como los análisis de línea y de borde para la segmentación de imágenes en citología y micrografías para la detección de la superficie de ventrículo izquierdo así como para el mejoramiento de muchas otras imágenes. Los coeficientes de compatibilidad de regiones proporcionan los medios para definir la cooperación deseada y la competencia entre puntos vecinos. La iteración del proceso permite que la información local se propague a través de la imagen para corregir los puntos que inicialmente parecen erróneamente clasificados

Su propuesta introduce el concepto de guías para manejar el proceso de relajación en la segmentación de imágenes. Las guías se usan con dos fines: para asignar el vector de probabilidad de una región inicial a cada punto de imagen y

para restringir la asignación de ciertas probabilidades y etiquetas de regiones en actualizaciones subsecuentes. Luego, se itera la relajación para actualizar los vectores y eventualmente llegar a la segmentación. El análisis de vertientes tiende a ser resistente al ruido pero requerir de computación intensiva. Como alternativa se puede operar sobre grupos de puntos en vez de hacerlo sobre puntos individuales. Esto puede hacerse formulando un método nuevo que incorpora el análisis de vertientes.

El análisis de vertientes ha sido usado como una solución para la segmentación de imágenes. Pensado originalmente dentro del contexto de una morfología matemática se ha expandido más allá. El análisis de vertientes empieza con una imagen de magnitud gradiente como entrada. Considera una imagen de entrada de una escala de gris bidimensional como una "superficie" topográfica tridimensional donde una imagen del valor escalar del punto de gris actúa como altura de superficie (el análisis de vertientes puede ser generalizado a imágenes en tres dimensiones, pero las analogías físicas son más complicadas en las dimensiones más altas, así que sólo el caso bidimensional se usará en este trabajo).

Las áreas de alto gradiente que corresponden a la región de los bordes aparecen como cadenas montañosas o líneas de vertiente. Las áreas de gradiente bajo que corresponden a las regiones interiores, aparecen como valles planos o cauces. El análisis de vertiente subdivide la imagen en cauces flanqueados por líneas de vertiente. Los cauces son las regiones de la imagen y las líneas de vertiente son los bordes de la región.

Si se implementan durante operaciones morfológicas estándar, los análisis de vertiente son computacionalmente muy intensos; sin embargo, se ha propuesto un método rápido basado en el concepto de una simulación de inmersión. El análisis de vertientes en su forma más básica tiende a ser sensible al ruido y a la sobresegmentación (se definen demasiadas regiones); las soluciones basadas en

marcadores de región de entrada y guías creadas por el operador han sido propuestas para aminorar estas dificultades. El método de etiquetado de relajación manejado por vertientes, usa la eficiencia computacional del análisis de vertientes y la resistencia al ruido del etiquetado de relajación.

Con una imagen de entrada dada, y un grupo de guías que definan la tarea de segmentación, el flujo básico del método es así: aplique el análisis de vertientes a la magnitud gradiente de la entrada para producir una sobresegmentación formada por muchos cauces asignando vectores iniciales de probabilidad a cada uno de los cauces, aplique el etiquetado de relajación a la representación de los cauces de la imagen para obtener la segmentación final.

Las guías influyen en la construcción de cauces y en la iniciación de la probabilidad del vector. Cabe hacer notar que la actualización de la segmentación opera ahora en una malla dispersa heterogénea y no lineal de cauces.

Queda, pues, descrito el método propuesto y en el siguiente capítulo se describe detalladamente una implantación del mismo y los resultados obtenidos de su aplicación, así como la comparación contra el método de Hsiao y Sawchuk (1989a y 1989b).

CAPITULO IV: IMPLANTACIÓN DEL MÉTODO Y RESULTADOS.

Para demostrar la utilidad del método propuesto se realizó una implantación del mismo. También se implantó el método de Hsiao y Sawchuk (1989a y 1989b), con la finalidad de comparar tanto el tiempo empleado por los métodos como la bondad de la segmentación obtenida. La implantación de esta propuesta se describe detalladamente en este capítulo y se compone de cinco etapas principales. El código en lenguaje de programación C puede encontrarse en el anexo 2.

4.1 PRIMERA: REALCE.

La energía de las texturas se estima inicialmente usando una ventana de tamaño pequeño para reducir la posibilidad de mezclar las estadísticas a lo largo de los bordes de la región. En este caso la ventana es de 5 x 5 pixeles. Esta parte consiste en realizar una convolución con cada una de las siguientes máscaras de Laws (1980) como se explicó en el capítulo anterior, en el apartado 3.1: E5L5 que resalta los bordes y el nivel de gris; R5R5 que se encarga de subrayar las ondulaciones; E5S5 para remarcar los bordes y las formas y L5S5 que enfatiza los niveles de gris y la forma. El producto son cuatro imágenes en cada una de las cuales se han resaltado algunas características. La ecuación 1 se usó para calcular la convolución. En este caso la función g correspondió a cada una de las cuatro máscaras propuestas por Laws (1980): E5L5, R5R5, E5S5 y L5R5. Para una mayor claridad, la forma de calcular la convolución se ilustra en la figura 4.1.

$$f \otimes g = \sum_{k=i-2}^{i+2} \sum_{l=j-2}^{j+2} f(k,l) * g(k,l) - f(i,j) * g(i,j) \quad [1]$$

a	b	c
d	e	f
g	h	i

A	B	C
D	E	F
G	H	I

$$\text{Convolución} = A*a + B*b + C*c + D*d + F*f + G*g + H*h + I*i$$

Figura 4.1 Convolución de dos matrices 3 x 3.

Una vez calculada la energía de la textura, lo que se tiene son cuatro imágenes en las cuales se han subrayado en la primera los bordes y el nivel de gris; en la segunda las ondulaciones; en la tercera los bordes y las formas y en la cuarta los niveles de gris y la forma. Esto permite hacer un filtrado más preciso en la siguiente etapa.

4.2 SEGUNDA: SUAVIZADO.

Los valores estimados de la energía de las texturas son suavizados entonces por un método de filtrado para reducir la variabilidad de las estimaciones mientras se retiene la exactitud de los bordes. Usando tamaños pequeños (7 x 7) es posible calcular las estadísticas con mayor precisión cerca de los bordes y en las regiones pequeñas, pero con tan pocos datos las estadísticas son más pobres que si se usa una ventana grande debido al menor número de puntos. Por otro lado, usando ventanas grandes (15 x 15) se tienen más puntos de datos para reducir la variabilidad estadística de los cálculos en las regiones grandes pero los resultados cerca de los bordes entre las texturas y en las regiones pequeñas son

escasos. Debido a este problema en el segundo paso se calcula la desviación estándar en ventanas de 7 x 7. Esto se hace para cada una de las cuatro imágenes generadas en el proceso anterior. Para el manejo de los bordes, lo que se hace es repetir el valor de los pixeles extremos de la imagen para hacer una especie de marco de 3 pixeles alrededor de la misma. Este marco permite hacer cálculos en todo el tamaño de la imagen, pues de no tenerlo sería necesario empezar a procesar tres pixeles hacia adentro en todas las direcciones. La ecuación 2 empleada es la siguiente:

$$s(i,j) = \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} |g(k,l)| \quad [2]$$

donde n vale 7 y g(k,l) son los pixeles de la imagen.

Lo que se tiene en este momento ya no son propiamente imágenes sino estadísticas de las cuatro imágenes con las características resaltadas en el paso anterior. Sin embargo como el filtrado se realizó en una ventana pequeña las estadísticas que se tienen son pobres, por lo que se procederá a filtrar en ventana 15 x 15 en el siguiente paso.

4.3 TERCERA: MEDIA CONTROLADA.

Posteriormente se calcula una media controlada en ventanas 15 x 15. En este caso se divide la ventana en cuatro cuadrantes de 7 x 7 donde no se toma en cuenta el mismo renglón y columna del pixel actual y se calcula la media (m_{wi}) y la desviación estándar (v_{wi}) para cada uno de los cuatro cuadrantes de la ventana. Se conserva la media del cuadrante con la menor desviación estándar correspondiente. Es decir, se obtienen las cuatro desviaciones estándar correspondientes a los cuatro cuadrantes y se elige el cuadrante con la menor

dispersión de los datos. A este cuadrante se le calcula la media y ese es el dato que se guarda para representar a esta ventana 15 x 15. Esto sirve para que si en uno o más de los cuadrantes se mezclan texturas no se tome esto como representativo de la vecindad del pixel. Nuevamente es necesario trabajar cada una de las cuatro salidas del paso anterior, correspondientes a los cuatro máscaras de Laws (1980). La forma en que se divide la ventana se muestra en la figura 4.2. Para el manejo de los bordes, lo que se hace es repetir el valor de los pixeles extremos de la imagen para hacer una especie de marco de 7 pixeles alrededor de la misma. Este marco permite hacer cálculos en todo el tamaño de la imagen, pues de no tenerlo sería necesario empezar a procesar tres pixeles hacia adentro en todas las direcciones. Las ecuaciones 3 a 10 son utilizadas para el cálculo de la media (m_{w1}) y la desviación estándar (v_{w1}).

$$m_{w1}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} s(i+k,j-l) \quad [3]$$

$$v_{w1}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} [s(i+k,j-l) - m_{w1}(i,j)]^2 \quad [4]$$

$$m_{w2}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} s(i-k,j-l) \quad [5]$$

$$v_{w2}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} [s(i-k,j-l) - m_{w2}(i,j)]^2 \quad [6]$$

$$m_{w3}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} s(i-k,j+l) \quad [7]$$

$$v_{w3}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} [s(i-k,j+l) - m_{w3}(i,j)]^2 \quad [8]$$

$$m_{w4}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} s(i+k,j+l) \quad [9]$$

$$v_{w4}(i,j) = \frac{1}{w^2} \sum_{k=0}^{w-1} \sum_{l=0}^{w-1} [s(i+k,j+l) - m_{w4}(i,j)]^2 \quad [10]$$

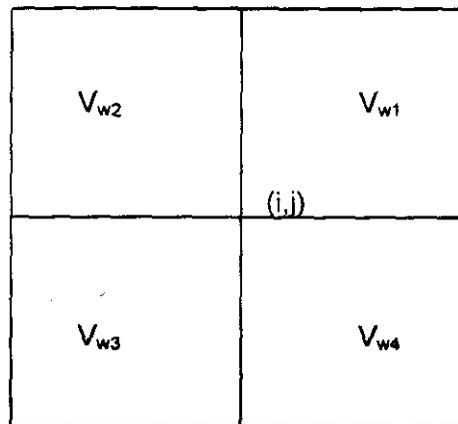


Figura 4.2 Modo de dividir la ventana 15 x 15.

4.4 CUARTA: AGRUPAMIENTO Y ETIQUETADO PROBABILISTICO INICIAL.

Los valores estimados son usados en un algoritmo de agrupamiento k-means para estimar las estadísticas de las clases. Las estadísticas estimadas de las clases son usadas para realizar un etiquetado probabilístico inicial. En esta parte finalmente se combina la información proveniente de las cuatro máscaras de

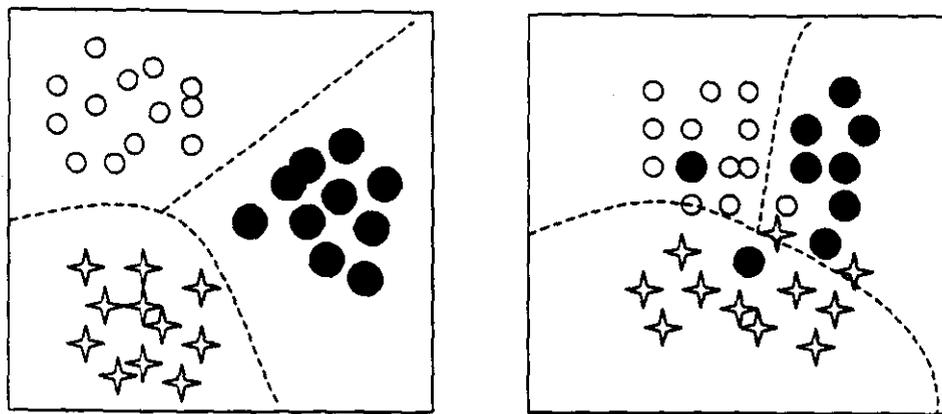
**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

Laws (1980) para que, utilizando toda la información se pueda realizar un buen etiquetado probabilístico inicial.

El parecido o similaridad entre píxeles o entre grupos de píxeles es un concepto fundamental que está detrás de los algoritmos de clasificación usados para el procesamiento de imágenes. Al agrupar datos con características similares se obtienen clases. Sin embargo, es importante determinar que tan bien corresponden las clases asignadas con las clases reales de la imagen que distinguen una categoría o zona de otra. En gran medida la capacidad para realizar una clasificación adecuada de una imagen se determina por la cantidad de traslape entre las características de las clases. Se considera que dos o más clases son muy separables si las particiones de las regiones de decisión pueden separarse sin error. Si los datos de dos clases marcadas como diferentes se traslapan, entonces es imposible lograr una frontera suave sin errores de clasificación, y se considera que la separabilidad es baja. Esto se ilustra en la figura 4.3

Tradicionalmente en geología, geografía y otras disciplinas, ha existido la noción de un mapa temático, que consiste en suponer que cada punto de la imagen puede ser etiquetado como perteneciente a una y solamente una categoría. Una división tan discreta es conveniente y atractiva por su simplicidad, pero no es particularmente cierta en lo que se refiere a las imágenes reales.

Muchos algoritmos de etiquetado producen una función de probabilidad para la asignación de un etiquetado de clase para cada píxel. Se produce una asignación dura si se selecciona la etiqueta de la clase que tiene la mayor probabilidad de ser correcta, en el campo de redes neuronales esto se conoce como el algoritmo de "el ganador toma todo" (WTA, por sus siglas en inglés). Por el contrario, si los valores de probabilidad se guardan, permitiendo etiquetas múltiples para cada píxel, se obtiene una clasificación suave. El espacio de decisión de los límites de una clasificación suave puede pensarse como difuso.



a)

b)

Figura 4.3 a) Datos con alta separabilidad. b) Datos con baja separabilidad.

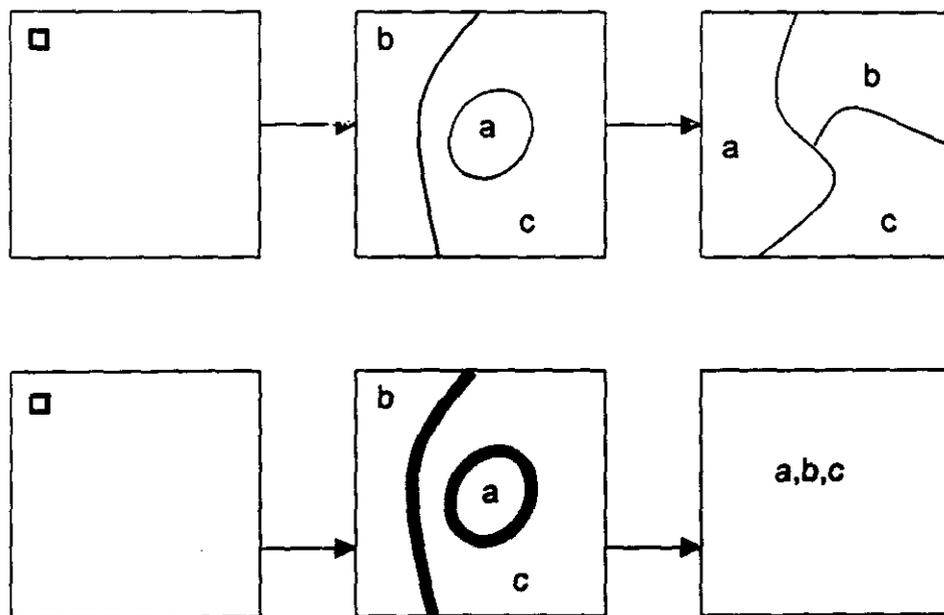


Figura 4.4 a) Clasificación dura b) Clasificación suave

En la figura 4.4 puede notarse la diferencia entre una clasificación suave y una dura. En ambos casos la imagen se pasa por el espacio de decisión para un etiquetado inicial. En la clasificación dura el "ganador toma todo", en donde sólo se permite una etiqueta para cada pixel. En la suave, la decisión tiene múltiples valores, con la posibilidad de más de una etiqueta por pixel. Cada etiqueta tiene una probabilidad asociada de ser correcta. Estas probabilidades pueden ser interpretadas de muchas formas una de las cuales es que indican la proporción de cada categoría dentro de un pixel.

Para el etiquetado probabilístico inicial se utilizó, además del agrupamiento realizado por k-means, una clasificación suave. Es por ello que el resultado de este proceso es una serie de probabilidades de pertenecer a cada clase por pixel.

4.5 QUINTA: RELAJACION.

Las restricciones espaciales se refuerzan a través del uso de un algoritmo de relajación, que se describe a continuación. Para la relajación la técnica usada fue la de vertientes y en particular se tomó la tabla de destino usada en este método. Esta consiste en examinar los ocho vecinos de un pixel y tomar en cuenta el etiquetado de estos vecinos para decidir el destino, es decir, la etiqueta que debe corresponder al pixel actual. En principio, si no se sabe a que clase corresponde un pixel pero todos o la mayoría de los que le rodean pertenece a una cierta clase, es razonable suponer que dicho pixel también debe pertenecer a esa clase. Por ejemplo un pixel de textura no identificada pero rodeado de pixeles de la clase 1 que puede corresponder a madera, seguramente será de esa misma clase y por tanto, madera

Para facilitar la identificación de las diferentes combinaciones posibles de ocho vecinos de un pixel se da un valor diferente que es una potencia de dos a cada posición de los ocho vecinos como se muestra en la figura 4.5. Dependiendo

de que la probabilidad de pertenecer a una clase de un pixel esté por encima de un nivel de corte se hace binaria la imagen y se multiplica cada vecino del pixel por su correspondiente en la matriz de destino y de aquí se obtiene un número entre 0 y 255. Se incluyen como anexo 1 las tablas de destino utilizadas para esta implantación.

8	4	2
16	?	1
32	64	128

Figura 4.5 Matriz de destino.

0	0	0
1	?	1
1	1	0

$$\text{Valor} = 1 \cdot 1 + 2 \cdot 0 + 4 \cdot 0 + 8 \cdot 0 + 16 \cdot 1 + 32 \cdot 1 + 64 \cdot 1 + 128 \cdot 0 = 112$$

Figura 4.6 Una configuración de vecinos que corresponde al número 112.

Estos cinco pasos componen el método propuesto y fueron implantados por separado en módulos independientes en lenguaje C. Los módulos son secuenciales en el orden en que fueron descritos y se relacionaron por medio de los archivos que eran la salida de uno constituían la entrada del siguiente. Los primeros cuatro pasos generan cada uno cuatro archivos para la siguiente etapa. El cuarto paso solamente genera un archivo para el último proceso. Esta

secuencia está ilustrada en la figura 4.7. No se unificaron todas las etapas en un solo sistema, puesto que el objetivo de la implantación era solamente demostrar el valor de la propuesta. El código fuente se encuentra en el anexo 2.

Siguiendo el método propuesto se realizaron nueve pruebas. Cinco de ellas sobre imágenes que contenían texturas artificiales, en las que se conocía exactamente donde debería realizarse la segmentación. Esto tuvo la ventaja de poder medir exactamente la efectividad del método. Estas imágenes y los resultados obtenidos por el método propuesto están en las figuras 4.8, 4.9, 4.10, 4.11 y 4.12. En esas mismas figuras se encuentran los resultados obtenidos al aplicar el método de Hsiao y Sawchuk (1989a y 1989b). Adicionalmente, se realizaron cuatro pruebas en imágenes reales, pero por cuestiones de tamaño de las mismas se tomó solamente una subimagen. Estas imágenes, los resultados del método propuesto y los resultados del método de Hsiao y Sawchuk (1989a y 1989b) se encuentran en las figuras 4.13, 4.14, 4.15 y 4.16. Por supuesto se marca en cada imagen qué subimagen se utilizó para la prueba.

Los resultados obtenidos con el método propuesto en las pruebas con imágenes artificiales fueron 98.3%, 98.6%, 99.1%, 99.45%, 99.7% de clasificaciones correctas. En todos los casos la clasificación fue espacialmente correcta, es decir no se presentaron "líneas rasgadas en los bordes" y el interior de las regiones fue sin agujeros, por lo que se considera que se cumplió el objetivo marcado. En lo que respecta a las imágenes reales se consigue buena definición en los bordes, pero por las características propias de las fronteras es razonable que a lo largo de estas las líneas no sean rectas. Tanto en el caso de las imágenes médicas como en el de las geográficas los usuarios consideraron la segmentación como muy adecuada. No se consideró conveniente cuantificar el porcentaje de clasificaciones correctas, por la dificultad que implica determinar con precisión el borde de segmentación.

Para validar el método se consideró importante hacer también una comparación en cuanto al tiempo con el método de Hsiao y Sawchuk (1989a y 1989b). Los resultados obtenidos se encuentran en la tabla 4.1. Cabe resaltar que el método propuesto aquí parece tomar un tiempo menor que el necesario para aplicar el método de Hsiao y Sawchuk (1989 a y b) cuya precisión espacial es menor o igual que la del método propuesto.

Imagen \ Método	1	2	3	4	5	6	7	8	9
Hsiao & Sawchuk	55	54	54	55	54	65	65	65	65
Método propuesto	51	49	50	51	50	59	59	58	59

Tabla 4.1 Tiempos (en segundos) comparativos de los diferentes métodos para cada una de las imágenes de prueba.

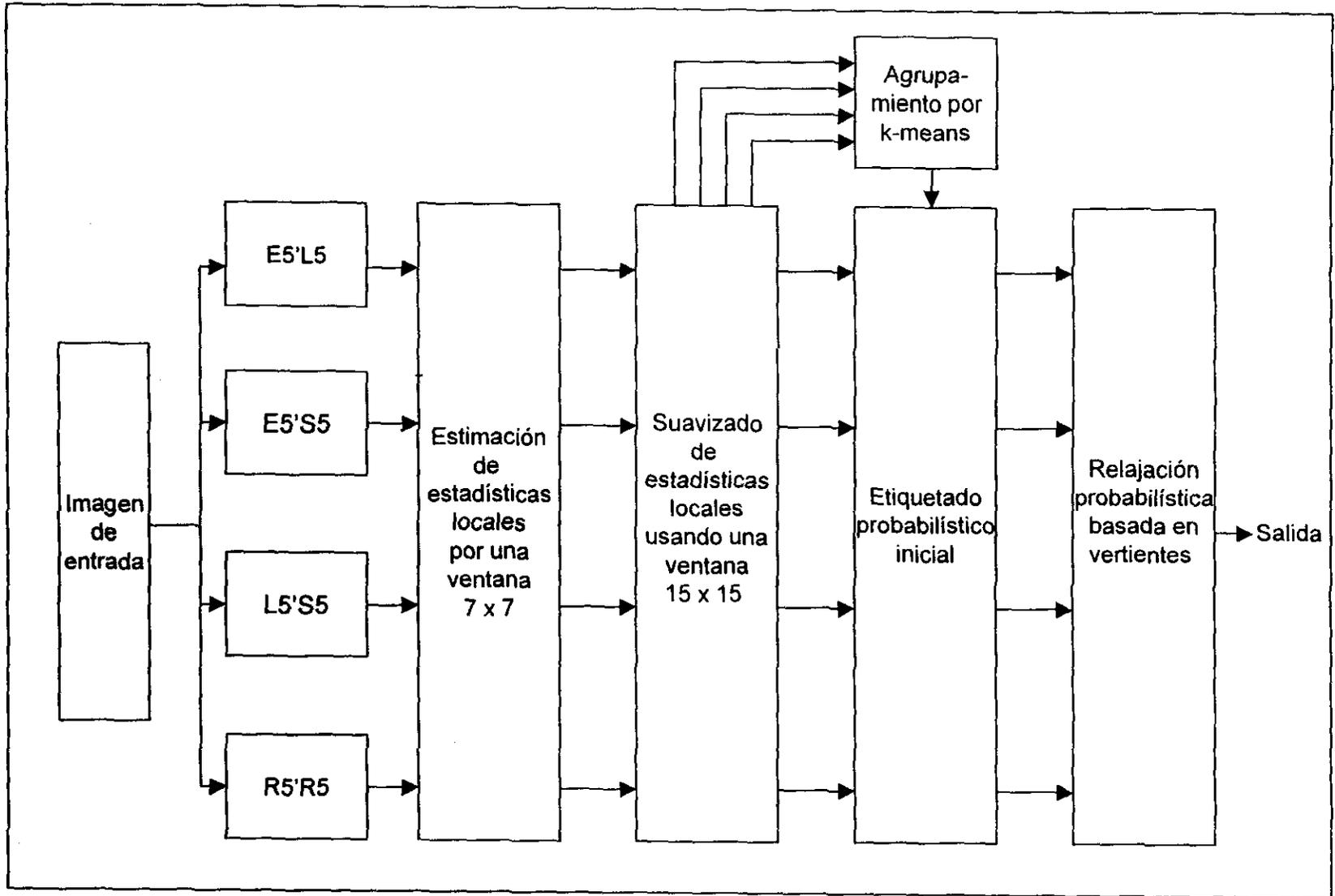


Figura 4.7 Diagrama de bloques del sistema de segmentación basado en texturas.

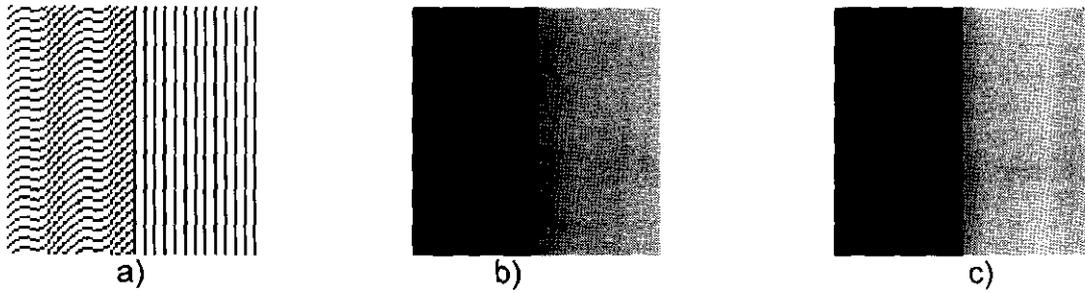


Figura 4.8 a) Imagen de prueba número 1. b) Resultado obtenido obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.

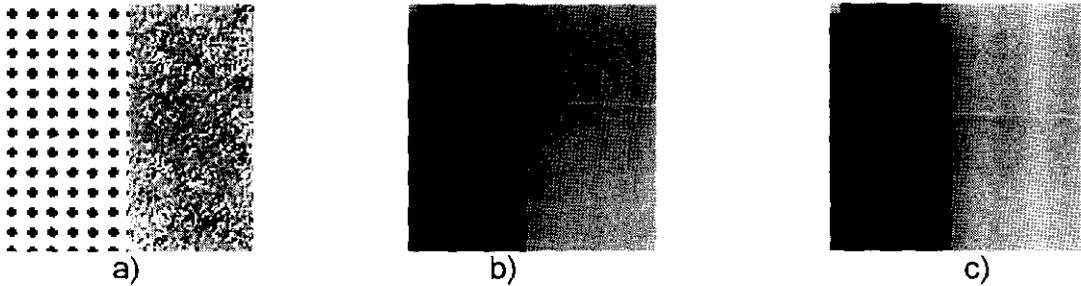


Figura 4.9 a) Imagen de prueba número 2. b) Resultado obtenido obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.

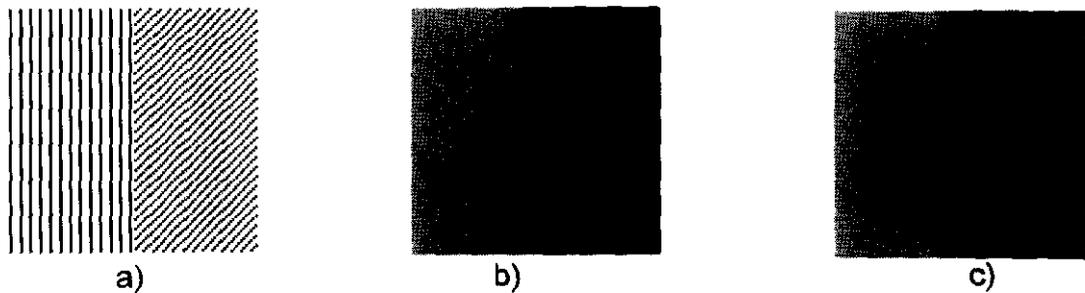


Figura 4.10 a) Imagen de prueba número 3. b) Resultado obtenido obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.

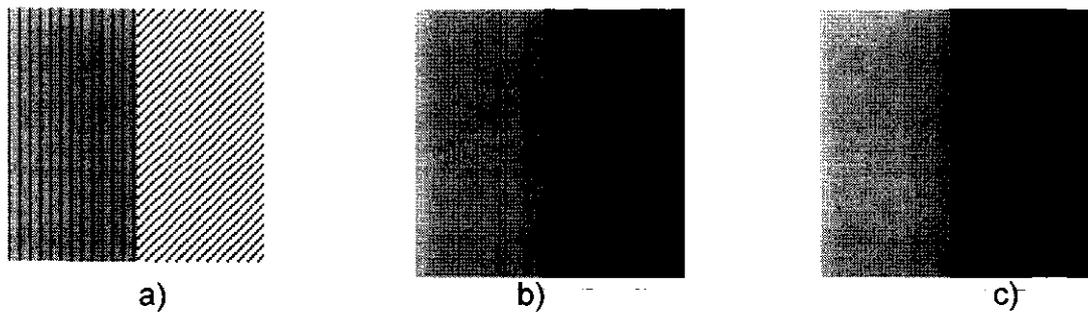


Figura 4.11 a) Imagen de prueba número 4. b) Resultado obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.

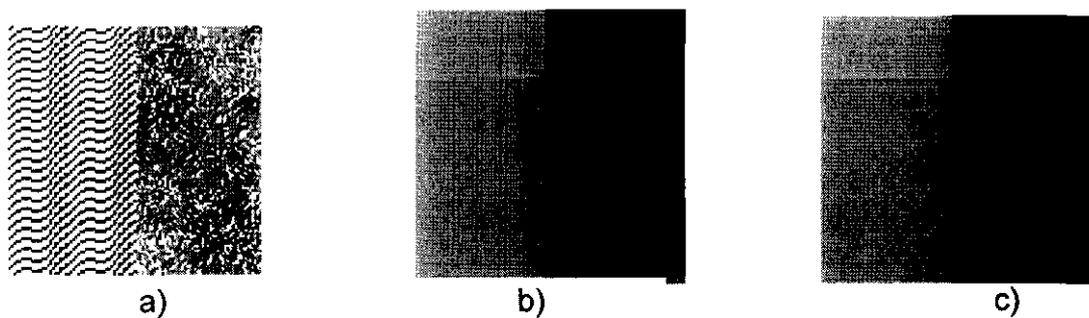
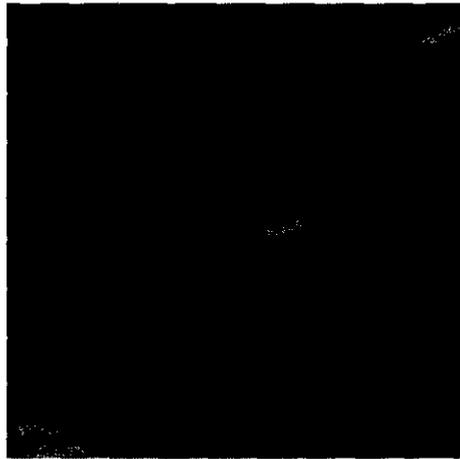


Figura 4.12 a) Imagen de prueba número 5. b) Resultado obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.



a)

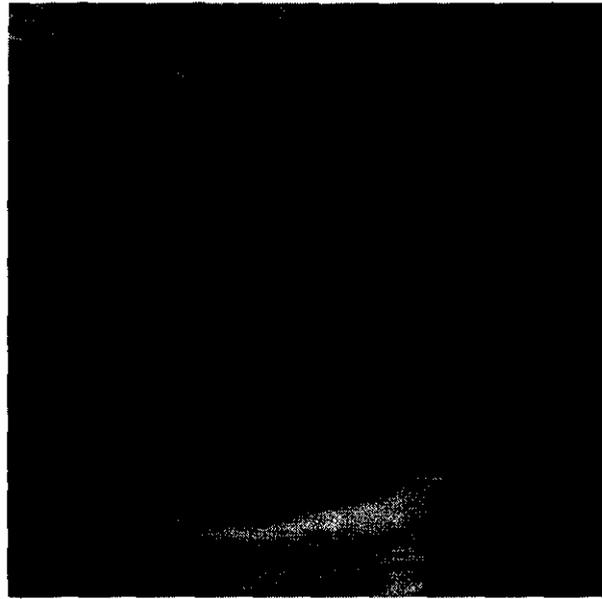


b)



c)

Figura 4.13 a) Imagen real de prueba número 1. Corresponde a un ultrasonido obstétrico. Se Marca la subimagen utilizada para la prueba. b) Resultado obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.



a)

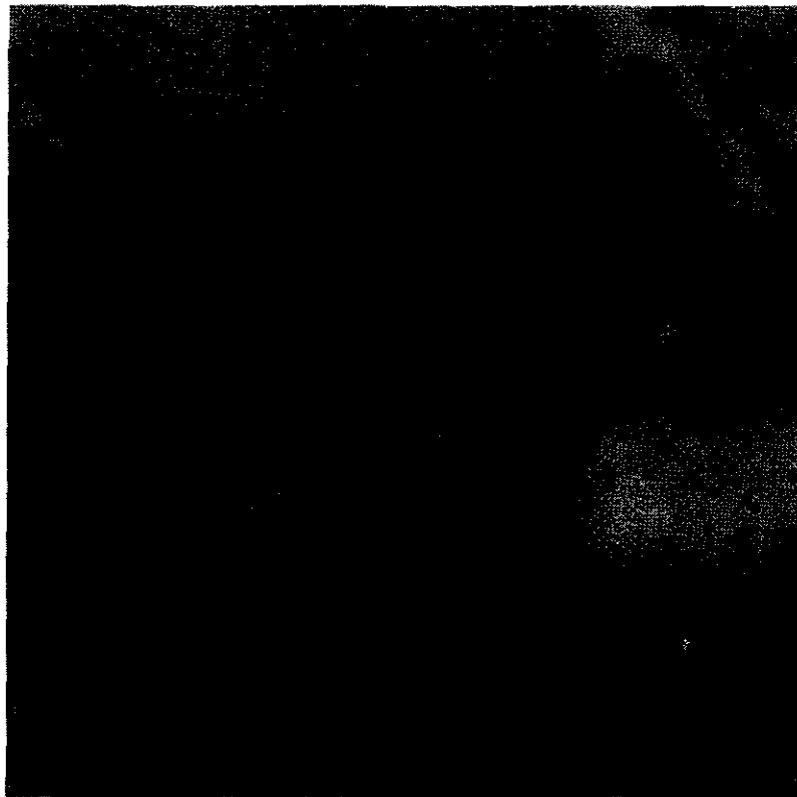


b)

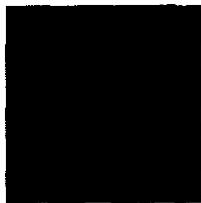


c)

Figura 4.14 a) Imagen real de prueba número 2. Corresponde a un ultrasonido obstétrico. Se Marca la subimagen utilizada para la prueba. b) Resultado obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.



a)



b)



c)

Figura 4.15 a) Imagen real de prueba número 3. Corresponde a una imagen satelital de Islandia. Se Marca la subimagen utilizada para la prueba. b) Resultado obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.



a)



b)



c)

Figura 4.16 a) Imagen real de prueba número 4. Corresponde a una imagen satelital de Inglaterra. Se Marca la subimagen utilizada para la prueba. b) Resultado obtenido por el método de Hsiao y Sawchuk. c) Resultado obtenido por el método propuesto.

CONCLUSIONES

Lo que se pretendía lograr al proponer un nuevo método para la segmentación basada en la textura era mejorar los resultados de la segmentación de imágenes con esta característica, especialmente a lo largo de las fronteras y mejorar la segmentación de los interiores de las regiones, tomando en cuenta las relaciones espaciales entre los píxeles.

El método planteado aquí combina varias propuestas anteriores: la energía de las texturas en la extracción de objetos, el concepto de suavizado adaptativo del ruido que toma en cuenta la naturaleza no estacionaria del problema y la relajación basada en el método de vertientes. Cumple ambos objetivos propuestos, como se mostró en el capítulo 4, con las pruebas y resultados de las figuras 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 y 4.16

Por supuesto no fue aleatoria la selección de cada uno de los métodos que se tomaron para combinarlos en esta propuesta, sino que se estudiaron las características de cada uno de los métodos para tomarse aquí y se hicieron pruebas para decidir qué características de cada uno de ellos favorece los objetivos propuestos en este trabajo. Cabe resaltar que a pesar de que se utilizaron métodos reportados en la literatura no necesariamente se usaron de la forma en que se encontraron en estos artículos, sino que parte de la propuesta presentada aquí es una nueva y muy diferente aplicación para métodos existentes y, por supuesto una forma distinta de combinarlos.

El método seguramente necesitará adaptaciones para ser utilizado con éxito en diferentes áreas del procesamiento de imágenes, puesto que cada caso especial tiene sus propias características. Sin embargo en lo general su funcionamiento es muy bueno, con más del 99% de clasificaciones correctas, en promedio.

Es muy claro que es necesario extender el método para trabajar con imágenes más grandes. Las pruebas que se han hecho hasta el momento han tenido muy buenos resultados y sirvieron para demostrar la viabilidad del método propuesto, no que la implantación sea correcta y completa. Por esto se proponen las siguientes recomendaciones:

1.- Sería conveniente realizar más pruebas, especialmente no utilizando partes de imágenes reales o texturas artificiales como se hizo aquí sino con imágenes médicas o geográficas completas, lo que implica que son de tamaño mayor.

2.- Es muy recomendable integrar esta serie de programas aislados en un sistema, que sea amigable al usuario y unificado. Al hacerlo se deberían incluir cuestiones como validar los datos de entrada y proporcionar ayudas al usuario.

3.- Es necesario hacer que la implantación sea más eficiente en cuanto al manejo de la memoria.

El método propuesto aquí tendrá el propósito de servir para dos grupos de trabajo, uno de ellos dedicado a información geográfica y el otro a imágenes medicas, en particular de ultrasonido, por lo que se considera que sobre todo el método será útil.

BIBLIOGRAFIA

Avilés C. (1998): "Utilidad de la Fusión de Datos en Segmentación de Áreas Texturadas", *Computación y Sistemas*. 3. pp 127-135.

Beucher S., Lantéjoul C. (1979): "Use of watersheds in contour detection", *Procedures of the International Workshop on Image Processing*, CCETT, Rennes, France.

Bolstein D., Ahuja N. (1989): "Shape from Texture: Integrating Texture-Element Extraction and Surface Estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp 1233-1251.

Bracewell R. (1965): *The Fourier Transform and Its Applications*. Mc Graw Hill, New York.

Caillol H., Pieczynski, Hillion A. (1997): "Estimation of Fuzzy Gaussian Mixture and Unsupervised Statistical Image Segmentation". *IEEE Transactions on Image Processing*, 6, pp 425-440.

Castleman K. R. (1996): *Digital Image Processing*. Prentice Hall, New Jersey.

Chang C., Chatterjee S. (1992): "A Hybrid Approach Toward Model-Based Texture Segmentation", *Pattern Recognition*, 25, pp 519-531.

Chen H., Dai Y. (1990): "Texture Segmentation by Least Squares Filters", *Signal Processing*, 19, pp 199-204.

Chellappa R. (1992): *Digital Image Processing*. Segunda Edición, IEEE Computer Society Press, Los Alamitos.

Cornelius T. L. (1998): *Image Processing and Pattern Recognition*. Academic Press, San Diego.

Davis L. S. (1981): "Image Texture Analysis Techniques Survey", *Digital Image Processing*, Editores J. C. Simon, R. M. Haralick, pp 189-201.

Davis L. S., Mitiche A. (1980): "Edge Detection in Textures", *Computer Graphics and Image Processing*, 12, pp 25-39.

Du Buf J. M. H., Kardan M., Spann M. (1990): "Texture Feature Performance for Image Segmentation", *Pattern Recognition*, 23, pp 291-309.

Duda, R. C., Hart, P. E. (1973): *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York.

Duff, M. J. B. (1987): *Intermediate Level Image Processing*. Academic Press, Inc., New York.

Earnshaw R, Vince J. and Jones H. (1997): *Visualization and Modeling*. Academic Press. San Diego.

Gambotto J.P. (1981): "Two Dimensional Time Series For Textures" *Digital Image Processing*, Editores J. C. Simon, R. M. Haralick, pp 211-229.

Geman D., Geman S. and Graffigne C. (1987): "Locating Texture and Object Boundaries", *Pattern Recognition Theory and Applications*. Editores P.A. Devijver, J. Kittler. Springer Verlag, Heidelberg, pp 1-8.

Geman S., Geman D. (1984): "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, pp 721-741.

González R. C., Woods R. E. (1992): *Tratamiento Digital de Imágenes*. Addison-Wesley / Díaz de Santos. Madrid.

Green, W. B. (1983): *Digital Image Processing: A Systems Approach* Van Nostrand Reinhold, New York.

Hansen M. W., Higgins W. E. (1997): "Relaxation Methods for Supervised Image Segmentation" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, pp 949-961.

Haralick R. M., Shapiro L. G. (1985): "Image Segmentation Techniques", *Computer Vision, Graphics and Image Processing*, 29, pp 100-132.

Haralick R. M., Shapiro L. G. (1992): *Computer and Robot Vision*. Addison-Wesley Publishing Company. Reading, MA.

Higuchi, L. T., Kawata, M. (1988): *Two-Dimensional Digital Signal Processing*. Marcel Dekker, Inc., New York.

Hord, R. M. (1986): *Remote Sensing: Methods and Applications*. John Wiley & Sons, Inc., New York.

Hsiao J. H., Sawchuk A. A. (1989a): "Supervised Textured Image Segmentation using Feature Smoothing and Probabilistic Relaxation Techniques",

IEEE. Transaction on Pattern Analysis and Machine Intelligence, 11, pp 1279-1292.

Hsiao J. H. Sawchuk A. A. (1989b): "Unsupervised Textured Image Segmentation using Feature Smoothing and Probabilistic Relaxation Techniques", *Computer Vision and Image Processing*, 46, pp 1-21.

Jain, A. K. (1988): *Fundamentals of Digital Image Processing*. Prentice Hall Inc., Eaglewood Cliffs.

Jain R., Kasturi R., and Schunck B. (1995): *Machine Vision*. Mc Graw Hill Series in computer science. New York.

James M. (1988): *Pattern Recognition*. Blackwell Scientific, Oxford, U. K., John Wiley & Sons.

Julesz B. (1981): "Textons, the Elements of Texture Perception and Their Interactions". *Nature*. 290 .pp 91-97.

Khotanzad A., Chen J. Y. (1989): "Unsupervised Segmentation of Textured Images by Edge Detection in Multidimensional Features", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11, pp 414-421.

Krishnamachari S., Chellappa R. (1997): "Multiresolution Gauss-Markov Random Field Models for Texture Segmentation". *IEEE Transactions on Image Processing*, 6, pp 251-267.

Lantejoul C., Beucher S. (1981): "On the use of the geodesic metric in image analysis", *Journal of Microscopy* 121, 39.

Laws, K.I. (1980): "Rapid Texture Identification" *Proceedings of the SPIE Conference on Image Processing for Missile Guidance*. pp 376-380.

Maravall D. (1994): *Reconocimiento de Formas y Visión Artificial*. Addison - Wesley Iberoamericana. Delaware.

Marroquin J. L. (1989): "A Markovian Random Field of Piecewise Straight Lines" *Biological Cybernetics*. pp 457-465.

Marroquin J. L. (1991): "Random Measure Fields and the Integration of Visual Information". *218th meeting of the Institute of Mathematical Statistics*.

Marroquin J. L. (1992): "Métodos Probabilísticos para el Proceso Digital de Imágenes". Comunicación personal.

Mitiche A., Davis L. S. (1981): "Optimal Edge Detection in Cellular Textures" *Digital Image Processing*, Editores J. C. Simon, R.M. Haralick, pp 155-176.

Modestino J. W., Fries R. W., Vickers L. (1980): "Stochastic Image Models Generated by Random Tessellations of the Plane", *Computer Graphics and Image Processing*, 12, pp 74-98.

Pavlidis, T. (1982): *Algorithms for Graphic and Image Processing*. Computer Science Press. Rockville, MD.

Pentland A. P. (1984): "Fractal Based Description of Natural Sciences". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6, pp 661-674.

Popat K., Picard R. W. (1997): "Cluster-Based Probability Model and Its Application to Image and Texture Processing". *IEEE Transactions on Image Processing*, 6, pp 268-284.

Pratt W. K. (1978): *Digital Image Processing*. John Wiley & Sons, Inc., New York.

Reed T. R., Wechsler, H. (1990): "Segmentation of Textured Images and Gestalt Organization using Spatial / Spatial-Frequency Representation" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, pp 1-12.

Reed T. R., Wechsler H., Werman M. (1990): "Texture Segmentation using a Diffusion Region Growing Technique", *Pattern Recognition*, 23, pp 953-960. 1990.

Rosenfeld A., Kak, A. C. (1982): *Digital picture processing*, Segunda Edición, 2 volúmenes, Academic Press, Inc., New York.

Rosenfeld A., Thurston M. (1971): "Edge and Curve Detection for Visual Scene Analysis", *IEEE Transactions on Computers*, 20, pp. 562-569.

Russ J. C. (1990): *Computer assisted microscopy: the measurement and analysis of images*. Plenum Press, New York.

Santiago J.L., López G., Márquez J., Corkidi G. (1992): "Una Implantación del Método de Vertientes para Segmentación de Imágenes". *VII Congreso Nacional de Instrumentación*. Xalapa, México.

Schalkoff, R. J. (1989): *Digital Image Processing and Computer Vision*. John Wiley & Sons, New York.

Schowengerdt R. A. (1997): *Remote Sensing: Models and Methods for Image Processing*. Segunda Edición. Academic Press. San Diego.

Sid-Ahmed M. A. (1995): *Image Processing.: Theory, Algorithms and Arquitectures* McGraw-Hill. New York.

Tou, J. T., Gonzalez, R. C. (1974): *Pattern Recognition Principles*. Addison Wesley Reading, MA.

Tuceryan M., Jain A. K. (1990): "Texture Segmentation using Voronoi Polygons", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp 717-728.

Unser M., Eden M. (1989): "Multiresolution Feature Extraction and Selection for Texture Segmentation", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11, pp 717-728.

Unser M., Eden M. (1990): "Nonlinear Operators for Improving Texture Segmentation and Features Extracted by Spatial Filtering", *IEEE Transactions on Systems, Man Cybernetics*, 20, pp 804-815.

Voorhees H., Poggio T. (1988): "Computing Texture Boundaries from Images" *Nature*, 333, pp 1-3.

Yaroslavsky, L .P. (1985): *Digital Picture Processing: An introduction* Springer-Verlag, Berlin.

Young, T. Y., Fu, K. S. (1986): *Handbook of Pattern Recognition and Image Processing*. Academic Press, San Diego.

configura.	decisión														
0	no	1 1	no	2 1	no	3 1	no	4	no	5 1	no	6	no	7 1	no
1 8	no	1 9 1	no	1 10 1	no	1 11 1	no	1 1 12	no	1 1 13 1	no	1 1 1 14	no	1 1 1 15 1	si
1 16	no	1 17 1	no	1 18 1	no	1 19 1	no	1 20	no	1 21 1	no	1 22	no	1 23 1	no
1 1 24	no	1 1 25 1	no	1 1 26 1	no	1 1 27 1	si	1 1 28	no	1 1 29 1	no	1 1 30	si	1 1 31 1	si
32 1	no	33 1 1	no	34 1 1	no	35 1 1	no	36 1 1	no	37 1 1	no	38 1 1 1	no	39 1 1 1	no
1 40 1	no	1 41 1 1	no	1 42 1 1	no	1 43 1 1	no	1 1 44 1	no	1 1 45 1 1	no	1 1 46 1 1	no	1 1 47 1 1	si
1 48 1	no	1 49 1 1	no	1 50 1 1	no	1 51 1 1	no	1 52 1	no	1 53 1 1	no	1 54 1 1	si	1 55 1 1	si
1 1 56 1	no	1 1 57 1 1	no	1 1 58 1 1	no	1 1 59 1 1	si	1 1 60 1	si	1 1 61 1 1	si	1 1 62 1 1	si	1 1 63 1 1	si


```

/* CONVOLUC.C Convolucion 5 x 5 */
/* Maria Guadalupe López Molina */
#include <conio.h>
#include <float.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#define m 100
#define n 100

void main (void) {
    int ven=2;
    int h, i, j, k, l, a, b;
    long sum;
    unsigned int lin [m+4] [n+4];
    FILE *arch, *arch1;
    static char w[4][5][5] = ( { { { 1,      -4,      6,      -4,      1},
                                {-4,      16,     -24,     16,     -4},
                                { 6,      -24,     36,    -24,     6},
                                {-4,      16,     -24,     16,     -4},
                                { 1,      -4,      6,      -4,      1 } } },
                                {{-1,      0,      2,      0,     -1},
                                 {-2,      0,      4,      0,     -2},
                                 { 0,      0,      0,      0,      0},
                                 { 2,      0,     -4,      0,      2},
                                 { 1,      0,     -2,      0,      1 } } },
                                {{-1,      0,      2,      0,     -1},
                                 {-4,      0,      8,      0,     -4},
                                 {-6,      0,     12,      0,     -6},
                                 {-4,      0,      8,      0,     -4},
                                 {-1,      0,      2,      0,     -1 } } },
                                {{-1,     -4,     -6,     -4,     -1},
                                 {-1,     -2,     -8,     -2,     -1},
                                 { 0,      0,      0,      0,      0},
                                 { 1,      2,      8,      2,      1},
                                 { 1,      4,      6,      4,      1 } } } );

    for (h=0; h<4; h++) { /* para cada filtro de Laws */
        if (( arch=fopen("prueba6.txt", "rt")) == NULL) {
            printf("Error al abrir el archivo de IMAGEN.\n");
            exit(-1);
        };
        switch (h) {
            case 0:
                if ((arch1= fopen("conv1", "wt"))==NULL) {
                    printf("Error al abrir el archivo CONV1. \n");
                    exit (-1);
                }
                break;
            case 1:
                if((arch1=fopen("conv2", "wt"))==NULL) {
                    printf("Error al abrir el archivo CONV2. \n");
                    exit (-1);
                }
        }
    }
}

```

```

        break;
    case 2:
        if((arch1=fopen("conv3", "wt"))==NULL) {
            printf("Error al abrir el archivo CONV3. \n");
            exit(-1);
        }
        break;
    case 3:
        if((arch1=fopen("conv4", "wt"))==NULL) {
            printf("Error al abrir el archivo CONV4. \n");
            exit(-1);
        }
    }
    for(i=ven;i<ven+m;i++){
        for(j=ven;j<ven+n;j++){
            fscanf(arch,"%d",&lin[i][j]);
        }
        for(j=0;j<ven;j++){
            lin[i][j]=lin[i][ven];
        }
        for(j=ven+n;j<2*ven+n;j++){
            lin[i][j]=lin[i][ven+n-1];
        }
    }
    for(j=0;j<2*ven+n;j++){
        for(i=0;i<ven;i++){
            lin[i][j]=lin[ven][j];
        }
        for(i=ven+m;i<2*ven+m;i++){
            lin[i][j]=lin[ven+m-1][j];
        }
    }
    for(i=ven; i<m+ven; i++) { /*para cada renglon de la imagen */
        for(j=ven; j<n+ven; j++) { /*para cada elemento del renglon i */
            sum=0;
            for(k=-ven; k<ven+1; k++)
                for(l=-ven; l<ven+1; l++) {
                    a= w[h][k+ven][l+ven];
                    b= lin[i+k][j+l];
                    sum+=(a * b);
                }
            fprintf(arch1,"%ld ",sum);
        }
        fprintf(arch1,"\n");
    }
    fclose(arch);
    fclose(arch1);
}

```

```

/* VENTANA7.C Desviación estandar en ventana 7 * 7 */
/* María Guadalupe López Molina */
#include <conio.h>
#include <float.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#define ven 3
#define n 100
#define m 100

void main(void) {
    int h, i, j, k, l;
    long sum;
    long lin[m+2*ven][n+2*ven];
    int fact=(2*ven+1)*(2*ven+1);
    FILE *arch, *arch1;
    for(h=0; h<4; h++) {          /* para cada imagen filtrada */
        switch (h) {
            case 0:
                if((arch1=fopen("desv1", "wt"))==NULL) {
                    printf("Error al abrir el archivo DESV1\n");
                    exit(-1);
                };
                if((arch=fopen("conv1", "rt"))==NULL) {
                    printf("Error al abrir el archivo CONV1.\n");
                    exit(-1);
                };
                break;
            case 1:
                if((arch1=fopen("desv2", "wt"))==NULL) {
                    printf("Error al abrir el archivo DESV2\n");
                    exit(-1);
                };
                if((arch=fopen("conv2", "rt"))==NULL) {
                    printf("Error al abrir el archivo CONV2.\n");
                    exit(-1);
                };
                break;
            case 2:
                if((arch1=fopen("desv3", "wt"))==NULL) {
                    printf("Error al abrir el archivo DESV3\n");
                    exit(-1);
                };
                if((arch=fopen("conv3", "rt"))==NULL) {
                    printf("Error al abrir el archivo CONV3.\n");
                    exit(-1);
                };
                break;
            case 3:
                if((arch1=fopen("desv4", "wt"))==NULL) {
                    printf("Error al abrir el archivo DESV4\n");
                    exit(-1);
                };
        }
    }
}

```

```

    if((arch=fopen("conv4", "rt"))==NULL) {
        printf("Error al abrir el archivo CONV4.\n");
        exit(-1);
    }
}
for(i=ven;i<ven+m;i++){
    for(j=ven;j<ven+n;j++){
        fscanf(arch,"%ld",&lin[i][j]);
    }
    for(j=0;j<ven;j++){
        lin[i][j]=lin[i][ven];
    }
    for(j=ven+n;j<2*ven+n;j++){
        lin[i][j]=lin[i][ven+n-1];
    }
}
for(j=0;j<2*ven+n;j++){
    for(i=0;i<ven;i++){
        lin[i][j]=lin[ven][j];
    }
    for(i=ven+m;i<2*ven+m;i++){
        lin[i][j]=lin[ven+m-1][j];
    }
}
for(i=ven; i<m+ven; i++) { /*para cada renglon de la imagen*/
    for(j=ven; j<n+ven; j++) { /*para cada elemento del renglon i*/
        sum=0;
        for(k=i-ven; k<i+ven+1; k++)
            for(l=j-ven; l<j+ven+1; l++)
                sum+=fabs(lin[k][l]);
        sum/=fact;
        fprintf(arch1,"%ld ",sum);
    }
    fprintf(arch1,"\n");
}
fclose(arch);
fclose(arch1);
}

```

```

/* VENTANA15.C Media controlada en ventana 15 * 15 */
/* Maria Guadalupe López Molina */
#include <conio.h>
#include <float.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#define lim 4
#define ven 8
#define n 100
#define m 100

void main(void) {
    int h,i,j,k,l,pos;
    long mw[lim],vw[lim],menor;
    long lin[m+2*ven][n+2*ven];
    FILE *arch, *arch1;
    int fact=lim*lim;
    for(h=0; h<4; h++) {          /* para cada imagen filtrada */
        switch (h) {
            case 0:
                if((arch1=fopen("media1", "wt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA1\n");
                    exit(-1);
                };
                if((arch=fopen("desv1", "rt"))==NULL) {
                    printf("Error al abrir el archivo DESV1.\n");
                    exit(-1);
                };
                break;
            case 1:
                if((arch1=fopen("media2", "wt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA2\n");
                    exit(-1);
                };
                if((arch=fopen("desv2", "rt"))==NULL) {
                    printf("Error al abrir el archivo DESV2.\n");
                    exit(-1);
                };
                break;
            case 2:
                if((arch1=fopen("media3", "wt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA3\n");
                    exit(-1);
                };
                if((arch=fopen("desv3", "rt"))==NULL) {
                    printf("Error al abrir el archivo DESV3.\n");
                    exit(-1);
                };
                break;
            case 3:
                if((arch1=fopen("media4", "wt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA4\n");
                    exit(-1);
                };
        }
    }
}

```

```

};
if((arch=fopen("desv4", "rt"))==NULL) {
    printf("Error al abrir el archivo DESV4.\n");
    exit(-1);
};
}
for(i=ven;i<ven+m;i++){
    for(j=ven;j<ven+n;j++){
        fscanf(arch,"%ld",&lin[i][j]);
        for(j=0;j<ven;j++){
            lin[i][j]=lin[i][ven];
        }
        for(j=ven+n;j<2*ven+n;j++){
            lin[i][j]=lin[i][ven+n-1];
        }
    }
    for(j=0;j<2*ven+n;j++){
        for(i=0;i<ven;i++){
            lin[i][j]=lin[ven][j];
        }
        for(i=ven+m;i<2*ven+m;i++){
            lin[i][j]=lin[ven+m-1][j];
        }
    }
    for(i=ven; i<ven+m; i++) { /*para cada renglon de la imagen*/
        for(j=ven; j<ven+n; j++) { /*para cada elemento del renglon i*/
            for(k=0; k<lim; k++)
                mw[k]=0;
            for(k=0; k<ven; k++)
                for(l=0; l<ven; l++) {
                    mw[0] +=lin[i+k][j-1];
                    mw[1] +=lin[i-k][j-1];
                    mw[2] +=lin[i-k][j+1];
                    mw[3] +=lin[i+k][j+1];
                }
            for(k=0; k<lim; k++) {
                mw[k] /=fact;
                vw[k]=0;
            }
            for(k=0; k<ven; k++)
                for(l=0; l<ven; l++) {
                    vw[0] +=(lin[i+k][j-1]-mw[0])*(lin[i+k][j-1]-mw[0]);
                    vw[1] +=(lin[i-k][j-1]-mw[1])*(lin[i-k][j-1]-mw[1]);
                    vw[2] +=(lin[i-k][j+1]-mw[2])*(lin[i-k][j+1]-mw[2]);
                    vw[3] +=(lin[i+k][j+1]-mw[3])*(lin[i+k][j+1]-mw[3]);
                }
            for(k=0; k<lim; k++)
                vw[k] /=fact;
            menor=vw[0];
            pos=0;
            for(k=1; k<lim; k++)
                if(vw[k] < menor) {
                    menor=vw[k];
                    pos=k;
                }
            fprintf(arch1,"%ld",mw[pos]);
        }
    }
    fprintf(arch1,"\n");
}

```

```

/* KMEANS.C algoritmos de etiquetado y relajacion */
/* María Guadalupe López Molina */
#include <conio.h>
#include <float.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#define m 625
#define n 25
#define num 2

void main(void){
    int pos,h,i,j,it,cuenta[20];
    float cota,min,mayor,menor;
    float nulo,tramo;
    unsigned char par[4][m],suma;
    float lin[m],cent[num];
    FILE *arch,*arch1;
    if ((arch1=fopen("kmean", "wt"))==NULL) {
        printf("Error al abrir el archivo KMEAN.\n");
        exit(-1);
    }
    for (h=0; h<4; h++) {          /* para cada filtro de Laws */
        switch (h) {
            case 0:
                if ((arch=fopen("media1", "rt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA1.\n");
                    exit(-1);
                }
                break;
            case 1:
                if ((arch=fopen("media2", "rt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA2.\n");
                    exit(-1);
                }
                break;
            case 2:
                if ((arch=fopen("media3", "rt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA3.\n");
                    exit(-1);
                }
                break;
            case 3:
                if ((arch=fopen("media4", "rt"))==NULL) {
                    printf("Error al abrir el archivo MEDIA4.\n");
                    exit(-1);
                }
        }
        it=0;
        for (i=0; i<n; i++) {
            for (j=0; j<n;j++) {
                fscanf(arch,"%e %e %e %e",&lin[it],&nulo,&nulo,&nulo);
                it++;
            }
        }
    }
}

```

```

    for (j=0; j<3*n;j++)
        fscanf(arch, "%e %e %e %e", &nulo, &nulo, &nulo, &nulo);
    }
    mayor=lin[0];
    menor=lin[0];
    for (i=1; i<m; i++) {
        if (lin[i]>mayor)
            mayor=lin[i];
        if (lin[i]<menor)
            menor=lin[i];
    }
    if (mayor!=menor)
        tramo=(mayor-menor)/20.0;
    else
        tramo=0;
    for (j=0; j<20; j++)
        cuenta[j]=0;
    for (i=0; i<m; i++)
        for (j=0; j<20; j++)
            if ((lin[i]>=(menor+tramo*j)) && (lin[i]<=(menor+(tramo*(j+1)))))
                cuenta[j]++;
    min=cuenta[0];
    pos=0;
    for (j=0; j<20; j++)
        if (min>cuenta[j]) {
            min=cuenta[j];
            pos=j;
        }
    for (i=0; i<m; i++)
        if (lin[i]<menor+tramo*pos)
            par[h][i]=0;
        else
            par[n][i]=1;
    fclose(arch);
}
for (i=0; i<m; i++) {
    suma=0;
    for (h=0; h<4; h++)
        if (par[h][i]==0)
            suma+=25;
    fprintf(arch1, "%d ", suma);
    if ((i % 25) == 24)
        fprintf(arch1, "\n");
}
fclose(arch);
}

```