

**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**ADQUISICIÓN Y RECONOCIMIENTO DE  
IMÁGENES**

**INFORME DE SUFICIENCIA**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ELECTRÓNICO**

**PRESENTADO POR:**

**JOSÉ CARLOS NAVARRO LÉVANO**

**PROMOCIÓN  
-1979-II**

**LIMA – PERÚ  
2006**

# **ADQUISICIÓN Y RECONOCIMIENTO DE IMÁGENES**

**A mis padres  
Carlos Alfredo Navarro Matheus y  
Esperanza Salome Lévano Montoya  
por sus enseñanzas y ejemplos**

## SUMARIO

El propósito del informe es presentar los tópicos y técnicas utilizadas en el tratamiento digital de las imágenes, así como sus aplicaciones utilizando el software MATLAB.

El interés de este tipo de tratamiento se deriva de la necesidad de mejorar la información pictórica para la interpretación humana y el procesamiento de datos para la percepción autónoma por una maquina.

El primer capítulo trata sobre una descripción general de los elementos más importantes de un sistema de adquisición y reconocimiento de imágenes. El segundo capítulo detalla el modelo matemático utilizado para la adquisición de imágenes y los medios mas usuales por los que se realiza la adquisición. El tercer capítulo describe el proceso matemático y estadístico para la realización del reconocimiento e interpretación de imágenes haciendo énfasis en los diversos modelos que se utilizan en la actualidad. El cuarto y quinto capítulos presentan las aplicaciones del software para el proceso de imágenes, MATLAB.

Se ha recopilado información de las mas usuales y modernas técnicas que se vienen utilizando en la actualidad y su aplicación mediante el software existente, debiendo precisarse que el procesamiento de imágenes va mejorando gracias al desarrollo de computadoras cada vez mas veloces, con alta capacidad de procesamiento y precios cada vez mas competitivos, con mayores capacidades de almacenamiento de información y la existencia de software amigable para el procesamiento como el MATLAB.

## ÍNDICE

### CAPITULO I

<b>DESCRIPCIÓN GENERAL</b>	<b>2</b>
1.1 Fundamentos	2
1.2 Representación digital de imágenes	3
1.3 Etapas fundamentales del procesamiento de imágenes	4
1.4 Elementos de los sistemas de procesamiento digital de imágenes	5
1.4.1 Adquisición de imágenes	5
1.4.2 Almacenamiento	8
1.4.3 Procesamiento	9
1.4.4 Comunicación	9
1.4.5 Presentación	10

### CAPITULO II

<b>ADQUISICIÓN DE LA IMAGEN DIGITAL</b>	<b>12</b>
2.1 Modelo de imagen simple	12
2.2 Muestreo y cuantificación	14
2.2.1 Muestro uniforme y cuantificación	15
2.2.2 Muestreo no uniforme y cuantificación	16
2.3. Relaciones básicas entre pixels	17

2.3.1 Vecinos de un píxel	18
2.3.2 Conectividad	28
2.3.3 Etiquetado de componentes conexos	20
2.4 Geometría de la imagen	21
2.4.1 Algunas transformaciones básicas	21
2.5 La película fotográfica	25
2.5.1 La estructura de la película y la exposición	25
2.5.2 Características de la película	28
<b>CAPÍTULO III</b>	
<b>RECONOCIMIENTO E INTERPRETACIÓN</b>	28
3.1 Elementos del análisis de imagen	29
3.2 Patrones y clases de patrones	30
3.3 Métodos de decisión teórica	31
3.4 Correspondencias	32
3.4.1 Clasificador de mínima distancia	32
3.4.2 Clasificadores estadísticos óptimos. Fundamentos	33
3.5 Redes Neuronales	36
3.6 Métodos estructurales	39
3.6.1 Correspondencia de números de forma	39
3.6.2 Correspondencia de cadenas	40
3.7 Métodos Sintácticos	41
3.8 Interpretación de imágenes	41
3.8.1 Fundamentos	42
3.8.2 Tipos de conocimientos	43
3.8.3 Sistemas lógicos	43

3.8.4 Redes semánticas	43
3.8.5 Sistemas Expertos de producción	44
<b>CAPITULO IV</b>	
<b>APLICACIONES EN MATLAB UTILIZANDO TOOLBOX</b>	<b>45</b>
4.1 Introducción al análisis y mejoramiento de imágenes	45
4.1.1 Los valores del píxel y estadísticas	45
4.1.2 Selección del píxel	46
4.1.3 El perfil de intensidad	47
4.1.4 Contorno de las imágenes	49
4.1.5 El Histograma de la imagen	50
4.2 Las estadísticas sumarias	51
4.2.1 Características de la medida	51
4.2.2 El análisis de la imagen	51
4.3 Mejorando la imagen	54
4.3.1 El ajuste de intensidad	54
4.3.2 Corrección Gamma	56
4.4 La eliminación del ruido	59
4.4.1 Filtrado adaptivo	61
<b>CAPITULO V</b>	
<b>APLICACIONES EN MATLAB PARA EL PROCESAMIENTO DE IMÁGENES</b>	<b>62</b>
5.1 Imágenes en Matlab	62
5.2 Almacenamiento y carga de imágenes en formato estándar	66
5.3 Exportación e importación de archivos de imágenes	
5.4 Manejo de elementos del procesamiento de imágenes	68

**CONCLUSIONES**

**73**

**BIBLIOGRAFÍA**

**74**

## PROLOGO

El propósito del informe es presentar los tópicos y técnicas utilizadas en el tratamiento digital de las imágenes, así como sus aplicaciones utilizando el software MATLAB.

El interés de este tipo de tratamiento se deriva de la necesidad de mejorar la información pictórica para la interpretación humana y el procesamiento de datos para la percepción autónoma por una maquina.

Se ha recopilado información de las mas usuales y modernas técnicas que se vienen utilizando en la actualidad y su aplicación mediante el software existente, debiendo precisarse que el procesamiento de imágenes va mejorando gracias al desarrollo de computadoras cada vez mas veloces, con alta capacidad de procesamiento y precios cada vez mas competitivos.

Deseo reconocer el apoyo del Ingeniero Jorge Del Carpio, al desarrollo del presente trabajo, pues gracias a la motivación que generó desde sus primeras exposiciones en las clases de procesamiento de señales en las aulas universitarias de la UNI, es que este trabajo ha podido ser culminado.

## **CAPITULO I DESCRIPCION GENERAL**

### **1.1. Fundamentos**

El interés por los métodos de tratamiento digital de imágenes deriva de dos áreas principales de aplicación: la mejora de la información pictórica para la interpretación humana y el procesamiento de los datos de la escena para la percepción autónoma por una máquina.

Las mejoras en los métodos de procesamiento para las imágenes digitales transmitidas se iniciaron en 1920, con la transmisión de fotografías y continuaron durante los siguientes treinta y cinco años. Sin embargo, fue el advenimiento combinado de las computadoras digitales de gran potencia y del programa espacial lo que puso de manifiesto el potencial de los conceptos del tratamiento digital de imágenes. La tarea de usar técnicas computacionales para mejorar las imágenes recibidas de una sonda espacial se inició en el Laboratorio de Propulsión Espacial (en Pasadena, California) en 1964 cuando las imágenes de la Luna transmitidas por el Ranger 7 fueron procesadas por un ordenador para corregir diversos tipos de distorsión de la imagen inherente a la cámara de televisión a bordo. Estas técnicas sirvieron como base para mejorar los métodos utilizados para realzar y restaurar las imágenes de la Luna enviadas por las misiones Surveyor, la serie de misiones Mariner dirigidas a Marte, los vuelos tripulados Apolo a la Luna, y otros.

### **1.2. Representación Digital de Imágenes**

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. La Figura 1 ilustra el convenio de ejes que se utiliza convencionalmente en los libros que tratan este tema.



Figura 1

A veces es útil representar una función imagen en perspectiva con un tercer eje que representa el brillo. Contemplada de esta forma, la Figura 1 aparecería como una serie de picos estrechos en las regiones con numerosos cambios en los niveles de brillo y otras regiones más suaves, o mesetas, donde los niveles de brillo varían poco o son constantes. Utilizando el convenio de asignar valores proporcionalmente más altos a las áreas más brillantes, se lograría que la altura de las diversas partes de esta representación fuese proporcional al correspondiente brillo de la imagen.

Una imagen digital es una imagen  $f(x,y)$  que se ha discretizado tanto en las coordenadas espaciales como en el brillo. Una imagen digital puede considerarse como una matriz cuyos índices de fila y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel gris en ese punto. Los elementos de una distribución digital de este tipo se denominan elementos de la imagen, o más comúnmente pixels o pels, abreviaturas de su denominación inglesa <<picture elements>>.

Aunque el tamaño de una imagen digital varía dependiendo de su aplicación, se demostrarán las grandes ventajas de seleccionar matrices cuadradas con tamaños y número de niveles de gris que sean potencias enteras de 2. Por ejemplo, un tamaño típico, comparable en calidad a una imagen monocroma de televisión, es una matriz de 512x512 puntos con 128 niveles de grises.

### 1.3. Etapas Fundamentales del Procesamiento de Imágenes

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 primera etapa del proceso es la **adquisición de la imagen** – es decir, 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. El sensor de imágenes puede ser también una cámara de barrido de líneas 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 imagen**. La función básica del preprocesamiento trata típicamente las técnicas de mejorar el contraste, eliminar el ruido y aislar regiones cuya textura indica la probabilidad de información alfanumérica.

La siguiente etapa trata de la **segmentación**. Definida de una forma general, la segmentación consiste en partir una imagen de entrada en sus partes constituyentes u objetos.

En general, la segmentación autónoma es una de las labores más difíciles del tratamiento digital de imágenes. Por una parte, un procedimiento de segmentación demasiado tosco dilata la solución satisfactoria de un problema de procesamiento de imágenes. Por otra, un algoritmo de segmentación débil o errática casi siempre garantiza que tarde o temprano habrá un fallo.

Por lo que se refiere al reconocimiento de caracteres, el papel fundamental de la segmentación es el de extraer caracteres individuales y palabras del fondo.

A la salida del proceso de segmentación habitualmente se tienen los datos de píxel en bruto, que constituyen bien el contorno de una región o bien 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 esquinas e inflexiones. La representación regional es 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 descripción**, también denominada selección de rasgos, consiste en extraer rasgos con alguna información cuantitativa de intereses o que sean fundamentales para diferenciar una clase de objetos de otra. En cuanto al reconocimiento de caracteres, descriptores tales como lagos (agujeros) y bahías proporcionan rasgos poderosos que ayudan a diferenciar una parte del alfabeto de otra.

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.

#### **1.4. Elementos de los Sistemas De Procesamiento Digital de Imágenes**

Los elementos de un sistema de propósito general capaz de realizar las operaciones de tratamiento digital de imágenes, que se acaban de presentar, se pueden agrupar de acuerdo a las funciones que este tipo de sistema generalmente lleva a cabo: 1) adquisición (equipos de video y scanners), 2) almacenamiento (discos ópticos, cinta magnética, cinta de video, discos magnéticos), 3) tratamiento (computadoras), 4) comunicación (computadoras) y 5) presentación de imágenes (monitores de televisión, pantallas de computadora, películas, impresoras).

##### **1.4.1 Adquisición de Imágenes**

Para la adquisición digital de imágenes 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. Otra amplia categoría de sensores está relacionada con la luz visible e infrarroja. Entre los dispositivos más frecuentemente empleados con este fin se encuentran los microdensitómetros, diseccionadores de imágenes, cámaras vidicom y matrices de detectores fotosensibles de estado sólido. El primer dispositivo necesita que la imagen a digitalizar esté en forma de transparencia (como un negativo de película) o de fotografía. Las cámaras vidicon y las matrices de elementos de estado de estado sólido pueden aceptar imágenes grabadas de esta forma y también pueden digitalizar imágenes naturales que posean el suficiente nivel de intensidad luminosa para excitar al detector.

En los microdensitómetros, la transparencia o fotografía se monta en un soporte plano o se arrolla alrededor de un tambor. El barrido se realiza dirigiendo un haz luminoso (que puede ser láser) sobre la imagen y desplazando el soporte o girando el tambor con relación al haz. En el caso de transparencias el haz pasa a través de la película, mientras que en el caso de fotografías se refleja por la superficie de la imagen. En ambos casos el haz se enfoca sobre un fotodetector y el nivel de gris de cada punto de la imagen es grabado por el detector en función de la intensidad del haz. Para obtener una imagen digital sólo se permiten en la salida valores discretos de la intensidad y de la posición. Aunque los microdensitómeros son dispositivos lentos, son capaces de llegar a elevados grados de precisión en la posición debido esencialmente a la naturaleza continua de la traslación mecánica empleada en el procesamiento de digitalización.

La operación de las cámaras de vidicon se basa en el principio de la fotoconductividad. Una imagen enfocada en la superficie de un tubo produce un patrón de conductividad variable que corresponde a la distribución de brillo en la imagen óptica. Un haz independiente de electrones, enfocado finamente, barre la superficie posterior del material fotoconductor y, por neutralización de la carga, crea una diferencia de potencial que produce en un colector una señal proporcional al patrón de brillo de la entrada. Cuantificando esta señal y la correspondiente posición del haz de barrido, se obtiene una imagen digitalizada.

Las matrices de elementos de estado sólido están compuestas por elementos discretos de silicio, denominados fotodetectores, que tienen un voltaje de salida proporcional a la intensidad de la radiación incidente. Las matrices de estado sólido están organizadas siguiendo una de las dos organizaciones geométricas principales: los sensores de barrido de línea y los sensores de área. Un sensor de barrido de línea está constituido por una fila de fotodetectores y produce una imagen bidimensional por el movimiento relativo entre la escena y el detector. Por ejemplo, los sensores de barrido de línea se utilizan ampliamente en los escáners de soporte plano. Un sensor de área está constituido por una matriz de fotodetectores y por ello es capaz de capturar una imagen de la misma forma que, por ejemplo, un tubo vidicon. Una de las grandes ventajas de las matrices de sensores de estado sólido es que pueden operar a velocidades muy altas (como, por ejemplo,  $1/10.000\text{seg}$ ), lo que los hace ideales en aplicaciones donde se necesite congelar el movimiento.

La tecnología utilizada en los sensores de estado sólido se basa principalmente en los dispositivos de carga acoplada (o CCD, de su nombre en inglés charge-coupled devices). Un CCD típico de barrido de línea contiene una fila de fotodetectores, dos puertas de transferencia, utilizadas para volcar el contenido de los elementos de la imagen en los denominados registros de transporte, y una puerta de salida, empleada para volcar el contenido de los registros de transporte en un amplificador, el cual proporciona una señal de salida proporcional al contenido de la fila de fotodetectores.

Las matrices de área de elementos de carga acoplada son similares a las de sensores de barrido de línea, excepto que aquí los fotodetectores están dispuestos en forma de matriz y se utiliza una combinación de registro de

transporte/puerta para separar columnas de fotodetectores. Los contenidos de los fotodetectores con numeración impar se llevan secuencialmente a los registros de transporte vertical y de ahí al registro de transporte horizontal. El contenido de este registro se introduce en un amplificador cuya salida es una línea de vídeo. Repitiendo este proceso para las líneas pares se completa el segundo campo de una imagen de televisión entrelazada. El barrido se repite 30 veces por segundo.

Los sensores de barrido de línea con resoluciones desde 256 hasta 4.096 elementos son muy comunes, mientras que la resolución de los sensores de área varía entre los 32x32 elementos del extremo inferior a los 256x256 elementos de un sensor de resolución media. También existen dispositivos de mayor resolución con 640x480 elementos, disponibles comercialmente a precios relativamente altos pero a menudo justificables. A precios mayores existen sensores de gran resolución que utilizan el movimiento mecánico de un chip. CCD para alcanzar los 2.048x2.048 elementos. Las matrices de CCD están normalmente empaquetadas como en las cámaras de televisión. La digitalización de la imagen se logra al enviar la señal de salida de vídeo de estas cámaras a un digitalizador, como se ha mencionado anteriormente.

#### **1.4.2 Almacenamiento**

Una imagen de 8bits y 1.024 x 1.024 pixels 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 tratamiento 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.

El tamaño de la información almacenada se mide en bytes (8 bits), Kbytes (kilobytes, o miles de bytes), Mbytes (megabytes, o millones de bytes), Gbytes (gigabytes, o miles de millones de bytes) y Tbytes (terabytes, o billones de bytes).

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 aproximación (zoom) prácticamente instantánea de la imagen, así como desplazamientos verticales (scroll) y horizontales (pan). La capacidad de almacenamiento en una tarjeta y por la densidad de almacenamiento de los elementos de memoria utilizados. Desde hace algún tiempo no es extraño disponer de 32 Mbytes de almacenamiento en una sola tarjeta de memoria temporal.

### **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 tratamiento 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. Por ejemplo, una aplicación importante del tratamiento digital es la microscopía con baja iluminación. Para reducir el ruido de las imágenes se necesita hacer el promedio de numerosas imágenes a la misma velocidad que se van adquiriendo (30 imágenes por segundo en la mayoría de los casos). La arquitectura en bus en todas las computadoras, excepto en unas pocas de altas prestaciones, no puede emplear la velocidad de datos necesaria para realizar esta operación. Así los sistemas actuales de procesamiento de imágenes son una combinación de computadoras, periféricos y hardware especializado en tratamiento de imágenes, de forma que la operación global está controlada por el software que se ejecuta en la computadora principal.

### **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 transición 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 la mayor parte de los libros sobre redes de computadoras explican los protocolos de comunicación más usuales.

La comunicación a través de grandes distancias representa un reto mucho más serio, especialmente cuando se trata de comunicar datos de una imagen, en lugar de resultados abstractos. Como podemos darnos cuentas las imágenes digitales contienen una cantidad importante de datos. Una línea telefónica de audio puede llegar a transmitir un máximo de 9.600 bits/seg.; así, para transmitir una imagen de 8 bits y 512x512 puntos a esta velocidad se necesitarían unos cinco minutos. Las uniones inalámbricas que emplean estaciones intermedias, como satélites, son mucho más veloces, pero también cuestan más. El resultado es que la transmisión de imágenes completas a largas distancias es un problema nada trivial. En ese sentido, las técnicas de compresión y descompresión de imágenes desempeñan un papel principal en el enfoque de este problema.

#### **1.4.5 Presentación**

Los monitores de televisión monocromos, y de color, son los principales dispositivos de presentación utilizados en los sistemas modernos de procesamiento de imágenes. Los monitores están gobernados por la salida (o salidas) de una placa de hardware de visualización de imágenes, que puede estar situada en el panel posterior de la computadora principal o bien ser parte del hardware asociado a un procesador 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. Otros medios de visualización incluyen tubos de rayos catódicos de acceso aleatorio (CRT) y dispositivos de impresión.

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 simple programa y relativamente pocos caracteres.

## CAPITULO II ADQUISICION DE LA IMAGEN DIGITAL

### 2.1. Modelo de Imagen Simple

El término imagen se refiere a una función bidimensional de la luz y la intensidad, a la que indicamos por  $f_{(x,y)}$ , donde el valor o amplitud de  $f$  en las coordenadas espaciales  $(x,y)$  da la intensidad (iluminación) de la imagen en este punto. Puesto que la luz es una forma de energía  $f_{(x,y)}$  debe de ser estrictamente mayor que cero y finita, es decir,

$$0 < f_{(x,y)} < \infty \quad (1)$$

Las imágenes que se perciben en las actividades visuales cotidianas provienen normalmente de la luz reflejada por los objetos. La naturaleza básica de  $f_{(x,y)}$  puede estar caracterizada por dos componentes:

1. La cantidad de luz incidente procedente de la fuente sobre la escena contemplada y
2. La cantidad de luz reflejada por los objetos de la escena.

De forma apropiada, reciben el nombre de componentes de iluminación y reflectancia, y se indican por  $i_{(x,y)}$  y  $r_{(x,y)}$  respectivamente. Las funciones  $i_{(x,y)}$  y  $r_{(x,y)}$  se combinan, como producto, para dar  $f_{(x,y)}$ :

$$f_{(x,y)} = i_{(x,y)} r_{(x,y)} \quad (2)$$

$$0 < i_{(x,y)} < \infty \quad (3)$$

$$0 < r_{(x,y)} < 1 \quad (4)$$

La ecuación (4) indica que la reflectancia está acotada entre 0 (absorción total) y 1 (reflexión total). La naturaleza de  $i_{(x,y)}$  está determinada por la fuente de luz y  $r_{(x,y)}$  está determinada por las características de los objetos de la escena.

Los valores dados en las ecuaciones (3) y (4) son límites teóricos. Los siguientes datos promedio ilustran algunos rangos típicos de  $i_{(x,y)}$ . En un día claro, el Sol puede producir más de 9.000cd (candelas)/pie<sup>2</sup> (1 pie~28cm) de iluminación sobre la superficie de la Tierra. Este valor decrece a menos de 1.000 cd/pie<sup>2</sup> en el caso de un día nublado. En una noche clara, la Luna llena produce una iluminación de alrededor de 0,01cd/pie<sup>2</sup>. El nivel típico de iluminación de una oficina comercial es de unas 100cd/pie<sup>2</sup>.

De forma similar, valores típicos de  $r_{(x,y)}$  son: 0,01 para terciopelo negro, 0,65 para el acero inoxidable, 0,80 para un muro con pintura completamente blanca, 0,90 para una lámina metálica plateada y 0,93 para la nieve.

La intensidad de una imagen monocromática  $f$  en las coordenadas  $(x,y)$  se la denominará nivel de gris ( $f$ ) de la imagen en este punto. De las ecuaciones (2) a la (4), resulta evidente que  $f$  está en el rango:

$$L_{\min} \leq f \leq L_{\max} \quad (5)$$

En teoría el único requisito sobre  $L_{\min}$  es que esa positiva y sobre  $L_{\max}$  que sea finita. En la práctica,  $L_{\min} = i_{\min} r_{\min}$  y  $L_{\max} = i_{\max} r_{\max}$ . Empleando como guía los valores precedentes de iluminación y reflectancia, se pueden esperar unos valores  $L_{\min} \approx 0,0005$  y  $L_{\max} \approx 100$  para aplicaciones de procesamiento de imágenes de interior.

Al intervalo  $[L_{\min}, L_{\max}]$  se lo denomina la escala de grises. Una práctica habitual consiste en desplazar numéricamente este intervalo hasta el intervalo  $[0,L]$ , donde en la escala,  $f = 0$  se considera como negro y  $f = L$  se considera como blanco. Todos los valores intermedios son tonos de gris que varían de forma continua entre el negro y el blanco.

## 2.2. Muestreo y Cuantificación

### 2.2.1 Muestreo Uniforme y Cuantificación

Para ser utilizable para el procesamiento por medio de computadora, una función de imagen  $f_{(x,y)}$  debe ser digitalizada tanto espacialmente como en su amplitud. La digitalización de las coordenadas espaciales  $(x,y)$  se denomina muestreo de la imagen y la digitalización de la amplitud se conoce bajo el nombre de cuantificación del nivel de gris.

Se supone que una imagen continua  $f_{(x,y)}$  se describe de forma aproximada por una serie de muestras igualmente espaciadas organizadas en forma de una matriz  $N \times M$  como se indica en la ecuación (1), donde cada elemento de la matriz es una cantidad discreta:

$$f_{(x,y)} \approx \begin{pmatrix} f_{(0,0)} & f_{(0,1)} \dots & f_{(0,M-1)} \\ f_{(1,0)} & f_{(1,1)} \dots & f_{(1,M-1)} \\ \vdots & \vdots & \vdots \\ f_{(N-1,0)} & f_{(N-1,1)} \dots & f_{(N-1,M-1)} \end{pmatrix} \quad (6)$$

El término de la derecha de la ecuación (6) representa lo que comúnmente se denomina una imagen digital. A cada elemento de la matriz se le puede denominar elemento de la imagen, **pixel** o **pel**, como se indicó anteriormente. Los términos imagen y pixels van a emplearse para indicar una imagen digital y sus elementos.

A veces puede resultar útil expresar el muestreo y la cuantificación en términos matemáticos más formales. Si  $Z$  y  $R$  representan a los conjuntos de números enteros y reales, respectivamente, el proceso de muestreo puede entenderse como una partición en una cuadrícula del plano  $xy$ , siendo las coordenadas del centro de cada elemento de la cuadrícula un par de elementos del producto cartesiano  $Z \times Z$  (también indicado por  $Z^2$ ), que es el conjunto de todos los pares ordenados de elementos  $(a,b)$  siendo  $a$  y  $b$  números enteros pertenecientes a  $Z$ . Por tanto  $f_{(x,y)}$  representa una imagen digital si  $(x,y)$  son enteros de  $Z \times Z$  y  $f$  es una función que asigna un nivel de gris (es decir, un número real del conjunto de los números reales  $R$ ) a cada par de coordenadas

(x,y) distinto. Esta asignación funcional es, obviamente, el proceso de cuantificación descrito anteriormente. Si los niveles de gris también son números enteros (como es el caso normal), entonces Z reemplaza a R y una imagen digital se convierte en una función bidimensional (2-D) cuyas coordenadas y valores de amplitud son números enteros.

Este proceso de digitalización necesita decisiones sobre los valores para N,M y el número de niveles de gris permitidos para cada pixel. En el procesamiento digital de imágenes es una práctica común que estas cantidades sean potencias enteras de 2; es decir:

$$N = 2^n, M = 2^k \quad (7)$$

$$G = 2^m \quad (8)$$

Donde G indica el número de niveles de gris. Se supondrá que los niveles discretos están igualmente espaciados entre 0 y L en la escala de grises. Empleando las ecuaciones (7) y (8) obtenemos el número, b, de bits necesarios para almacenar una imagen digitalizada:

$$b = N \times M \times m \quad (9)$$

Si  $M = N$

$$b = N^2 m \quad (10)$$

Por ejemplo, una imagen de 128x128 puntos con 64 niveles de gris necesita 98.304 bits de memoria. La tabla 1 resume los valores de b, de la ecuación (10), para algunos valores típicos de los parámetros N y m. Se obtiene el número correspondiente de bytes de 8 bits al hacerse la división entre 8.

Puesto que la ecuación (6) es una aproximación a una imagen continua, llegados a este punto es razonable preguntarse:

**¿Qué muestreo y cuántos niveles de gris son necesarios para una buena aproximación?**

La resolución (el grado de detalle discernible) de una imagen depende estrechamente de estos dos parámetros.

Cuanto más se incrementan más se aproxima la imagen digitalizada a la imagen original. Sin embargo, la ecuación (9) indica claramente la desafortunada circunstancia de que el almacenamiento y, consecuentemente, las necesidades de procesamiento crecen rápidamente en función de N, M y m.

A la vista de los comentarios precedentes – y suponiendo por conveniencia imágenes cuadradas - consideremos el efecto que tienen sobre la calidad de la imagen la variación de los parámetros N y m. Es difícil definir qué es una “buena imagen”, puesto que la calidad de la misma no sólo es muy subjetiva sino que también depende mucho de las necesidades de una aplicación determinada.

**Tabla 1: Número de bits de almacenamiento para varios valores de N y m**

$N \backslash m$	1	2	3	4	5	6	7	8
32	1.024	2.048	3.072	4.096	5.120	6.144	7.168	8.192
64	4.096	8.192	12.288	16.384	20.480	24.576	28.675	32.768
128	16,384	32.768	49.152	65.536	81.920	98.304	114.68	131.07
256	65.536	131.07	196.60	262.14	327.68	393.21	8	2
512	262.14	2	8	4	0	6	458.75	524.28
1024	4	524.28	786.43	1.048.5	1.310.7	1.572.8	2	8
	1.049.5	8	2	76	20	64	1.835.0	2.097.1
	76	2.097.1	3.145.7	4.194.3	5.242.8	6.291.4	08	52
		52	28	04	80	56	7.340.0	8.388.6
							32	08

### 2.2.2 Muestreo no Uniforme y Cuantificación

Para un valor fijo de la resolución espacial, la apariencia de una imagen puede mejorarse en muchos casos empleando un esquema adaptativo en el que el proceso de muestreo dependa de las características de la imagen. En general, se necesita un muestreo fino en las proximidades de las transiciones bruscas en los niveles de gris, mientras que se puede aplicar un muestreo tosco en las regiones relativamente suaves. Considérese, por ejemplo, una simple imagen de

una cara sobre un fondo uniforme. Es evidente que el fondo contiene muy poca información detallada y puede ser adecuadamente representado empleando un muestreo tosco. Sin embargo, la cara contiene muchos más detalles. Si las muestras que no se han empleado en el fondo, emplean para esta región de la imagen, el resultado global tiende a mejorar. En la distribución de las muestras, la mayor concentración debe emplearse en los contornos de transición entre niveles de gris; como, en este ejemplo, el contorno entre la cara y el fondo.

La necesidad de identificar los contornos, aunque solamente sea de forma aproximada, es uno de los inconvenientes definidos del muestreo no uniforme. Este método tampoco es práctico para el caso de imágenes que contengan regiones uniformes relativamente pequeñas. Por ejemplo, el muestreo no uniforme sería difícil de aplicar en la imagen de una multitud de personas densa.

Cuando el número de niveles de gris debe mantenerse reducido, el empleo de niveles desigualmente espaciados es a menudo deseable en el proceso de cuantificación. Un método similar a la técnica de muestreo no uniforme que se acaba de presentar puede emplearse para la distribución de niveles de gris en una imagen. Sin embargo, como el ojo es relativamente pobre estimando grados de grises cerca de cambios de nivel fuertes, la mejor aproximación en este caso consiste en emplear pocos niveles de gris cerca de los bordes. Los restantes niveles pueden emplearse en zonas donde las variaciones de los niveles de gris sean suaves, evitando o reduciendo así los falsos contornos que a menudo aparecen en estas zonas si también están cuantificadas toscamente.

Este método está sujeto a las observaciones precedentes sobre la detección de contornos y contenido en detalles. Una técnica alternativa que resulta particularmente atractiva para distribuir los niveles de gris consiste en calcular la frecuencia de aparición de todos los niveles permitidos. Si, en un cierto rango, algunos niveles de gris aparecen frecuentemente, mientras que otros aparecen raramente, los niveles de cuantificación deben estar finamente espaciados en este rango y ampliamente espaciados fuera de él. Este método se denomina a menudo cuantificación en bandas.

### **2.3. Relaciones Básicas Entre Pixels.**

Se considerarán algunas relaciones básicas pero importantes entre los pixels de una imagen digital. Como se mencionó anteriormente, una imagen se

indica por  $f_{(x,y)}$ . Cuando se haga referencia a un pixel en particular, se emplearán letras minúsculas, como  $p$  y  $q$ . Un subconjunto de pixels de  $f_{(x,y)}$  se indicará mediante  $S$ .

### 2.3.1 Vecinos de un Píxel

Un pixel  $p$  de coordenadas  $(x,y)$  tiene cuatro vecinos horizontales y verticales cuyas coordenadas vienen dadas por:

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

Este conjunto de pixels, 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 imagen digital si  $(x,y)$  está en el borde de la imagen.

Los cuatro vecinos en diagonal del  $p$  tienen las 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 de  $N_D(p)$  y  $N_8(p)$  caen fuera de la imagen si  $(x,y)$  está en el borde de la misma.

### 2.3.2. Conectividad

La conectividad entre pixels es un concepto importante empleado para establecer los límites de los objetos y los componentes de áreas en una imagen. Para determinar si dos pixels están conectados, debe determinarse si son adyacentes en algún sentido (como ser 4-vecinos) y si sus niveles de gris cumplen un criterio especificado de similitud (como ser iguales). Por ejemplo, en una imagen binaria con valores 0 y 1, dos pixels pueden ser 4-vecinos pero no estarán conectados a menos que tengan el mismo valor.

Sea  $V$  el conjunto de valores de nivel de gris empleados para definir la conectividad; por ejemplo, en una imagen binaria, se tendrá  $V=\{1\}$  para la conectividad entre pixels con valor 1. En una imagen con escala de grises, para

la conectividad entre pixels con un rango de valores de intensidad de, por ejemplo, 32 a 64, se tiene  $V = \{32, 33, \dots, 63, 64\}$ .

Se consideran tres tipos de conectividad:

a) 4-conectividad. Dos pixels  $p$  y  $q$  con valores dentro de  $V$  están 4-conectados si  $q$  pertenece a  $N_4(p)$ .

b) 8-conectividad. Dos pixels  $p$  y  $q$  con valores dentro de  $V$  están 8-conectados si  $q$  pertenece a  $N_8(p)$ .

c)  $m$ -conectividad (conectividad mixta). Dos pixels  $p$  y  $q$  con valores dentro de  $V$  están  $m$ -conectados sí:

i)  $q$  pertenece a  $N_4(p)$ , o bien

ii)  $q$  pertenece a  $N_D(p)$  y además el conjunto  $N_4(p) \cap N_4(q)$  es vacío.

(Este es el conjunto de pixels que son 4-vecinos de  $p$  y de  $q$  cuyos valores están en  $V$ ).

La conectividad mixta es una modificación de la 8-conectividad que se introdujo para eliminar los múltiples caminos de conexión que aparecen a menudo cuando se emplea la 8-conectividad. Por ejemplo, considérese la distribución de pixels mostrada en la Figura 2(a). Para  $V = \{1\}$ , los caminos entre 8-vecinos del pixel central se muestran con líneas continuas en la Figura 2(b). Obsérvese la ambigüedad de los caminos de conexión que resulta al permitir la 8-conectividad. Esta ambigüedad se elimina empleando la  $m$ -conectividad, como se muestra en la Figura 2(c).

```

0 1 1
0 1 0
0 0 1

```

(a)

```

0 1 1
0 1 0
0 0 1

```

(b)

```

0 1 1
0 1 0
0 0 1

```

(c)

**FIGURA 2**

(a)	Distribución de pixels
(b)	8-vecinos del pixel central
(c)	$m$ -vecinos del mismo pixel
Las líneas son caminos entre ese pixel y sus vecinos	

Un pixel  $p$  es adyacente de un pixel  $q$  si están conectados. Se puede definir 4-,8- o  $m$ -adyacencia, dependiendo del tipo de conectividad especificada. Dos subconjuntos de la imagen  $S_1$  y  $S_2$  son adyacentes si algún pixel de  $S_1$  es adyacente a algún pixel de  $S_2$ .

Un camino desde el pixel  $p$  de coordenadas  $(x,y)$  al pixel  $q$  de coordenadas  $(s,t)$  es una sucesión de diversos pixels de coordenadas:

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

Donde:

$(x_0, y_0) = (x,y)$  y  $(x_n, y_n) = (s,t)$ ,  $(x_i, y_i)$  es adyacente a  $(x_{i-1}, y_{i-1})$ ,  $1 \leq i \leq n$ , y  $n$  es la longitud del camino. Así podemos definir 4-,8- y  $m$ -caminos, dependiendo del tipo de adyacencia especificado.

Si  $p$  y  $q$  son pixels de un subconjunto  $S$  especificado de la imagen, se dirá que  $p$  está conectado con  $q$  dentro de  $S$  si existe un camino desde  $p$  hasta  $q$  que consista totalmente de pixels de  $S$ . Para cualquier pixel  $p$  dentro de  $S$ , el conjunto de pixels de  $S$  conectados a  $p$  se denomina componente conexa de  $S$ . Por tanto, cualquier par de pixels de una misma componente conexa están conectados entre sí, y componentes conexas distintas son disjuntas.

La capacidad de asignar etiquetas diferentes a las distintas componentes conexas disjuntas de una imagen es de importancia fundamental en el análisis automatizado de la imagen. En la sección siguiente se desarrollará un sencillo procedimiento secuencial de etiquetado de componentes conexas que actúa sobre filas de una imagen binaria cada vez.

### 2.3.3. Etiquetado de Componentes Conexas

Imagínese el barrido de una imagen pixel a pixel, de izquierda a derecha y de arriba abajo y supóngase que, por el momento, estamos interesados en componentes 4-conexas. Sea  $p$  el pixel en cada paso del proceso de barrido y sean  $r$  y  $t$  los vecinos superior e izquierdo de  $p$ , respectivamente.

La naturaleza de la secuencia de barrido asegura que cuando se llega a  $p$ , los puntos  $r$  y  $t$  ya han sido encontrados (y etiquetados sí fuesen unos 1). Con los conceptos anteriormente establecidos, considérese el procedimiento siguiente: si el valor de  $p$  es 0, simplemente se continúa hasta la siguiente

posición de barrido. Si el valor de  $p$  es 1, se examinan  $r$  y  $t$ . Si ambos son 0, se asigna una nueva etiqueta a  $p$  (por lo que hasta ahora se sabe, basados en la información presente, es la primera vez que se ha encontrado esta componente conexa). Si sólo uno de los dos vecinos es igual a 1, se asigna su etiqueta a  $p$ . Si ambos son 1 y tienen la misma etiqueta, se asigna una de las etiquetas a  $p$  y una nota de que ambas etiquetas son equivalentes (es decir, los puntos  $r$  y  $t$  están conectados a través de  $p$ ). Al final del barrido, todos los puntos con valor igual a 1 han sido etiquetados, aunque algunas de estas etiquetas puedan ser equivalentes. Todo lo que necesitamos hacer ahora es clasificar todos los pares de etiquetas equivalentes en clases de equivalencia, asignar una etiqueta diferente a cada clase, y luego dar una segunda pasada a través de la imagen reemplazando cada etiqueta por la etiqueta asignada a su clase de equivalencia.

Para etiquetar componentes 8-conectadas debemos proceder de la misma forma, pero ahora los vecinos diagonales superiores, indicados por  $q$  y  $s$ , también han de ser examinados. La naturaleza de la secuencia de barrido asegura que estos vecinos hayan sido ya procesados en el momento que se llegue al punto  $p$ . Si  $p$  es 0 será necesario moverse a la siguiente posición de barrido. Si  $p$  es 1 y los restantes cuatro vecinos son todos 0, ha de asignarse una nueva etiqueta a  $p$ . Si dos o más de los vecinos son 1, entonces ha de asignarse su etiqueta a  $p$ . Si dos o más de los vecinos son 1, entonces ha de asignarse una de sus etiquetas a  $p$  y ha de hacerse una anotación oportuna de las equivalencias. Tras completar el barrido de la imagen, se han de convertir los pares de etiquetas equivalentes en clases de equivalencia, asignando una única etiqueta para cada clase. Finalmente se realiza un segundo barrido de la imagen, cambiando cada etiqueta por la etiqueta asignada a su clase de equivalencia.

## **2.4. Geometría de la Imagen**

A continuación se presentan varias transformaciones importantes empleadas en el estudio de la imagen.

### **2.4.1. Algunas transformaciones Básicas**

Presentaremos el desarrollo de una representación única para problemas como la rotación de la imagen, el cambio de escala y la traslación. Todas las transformaciones están expresadas en un sistema de coordenadas cartesiano tridimensional (3-D), en el que las coordenadas de un punto genérico se

representan por  $(X,Y,Z)$ . En los casos que impliquen imágenes bidimensionales (2-D), se mantendrá la convención anterior de representación en minúsculas  $(x,y)$  para indicar las coordenadas de un pixel. Hacer referencia a  $(X,Y,X)$  como coordenadas globales de un punto es la terminología común.

#### 2.4.1.1. Traslación

Supóngase que se trata de trasladar un punto de coordenadas  $(X,Y,X)$  a un nuevo emplazamiento, empleando unos desplazamientos  $(X_o,Y_o,Z_o)$ . La traslación se realiza fácilmente empleando las ecuaciones:

$$\begin{aligned} X^* &= X + X_o \\ Y^* &= Y + Y_o \\ Z^* &= Z + Z_o \end{aligned} \quad (11)$$

Donde  $(X^*, Y^*, Z^*)$  son las coordenadas del nuevo punto. La ecuación (11) puede expresarse en forma matricial como:

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_o \\ 0 & 1 & 0 & Y_o \\ 0 & 0 & 1 & Z_o \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (12)$$

A menudo resulta provechoso concatenar varias transformaciones para producir un resultado compuesto, como una traslación, seguida de un cambio de escala y finalmente de una rotación. El empleo de matrices cuadradas simplifica considerablemente la notación necesaria para representar este proceso. Recordando esto, la ecuación (12) puede escribirse como:

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_o \\ 0 & 1 & 0 & Y_o \\ 0 & 0 & 1 & Z_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (13)$$

En términos de los valores de  $X^*, Y^*, Z^*$ , las ecuaciones (12) y (13) son equivalentes.

A lo largo de esta sección se empleará una representación matricial única,

$$\mathbf{V}^* = \mathbf{A}\mathbf{v} \quad (14)$$

donde  $\mathbf{A}$  es una matriz de transformación  $4 \times 4$ ,  $\mathbf{v}$  es un vector columna que contiene las coordenadas originales:

$$\mathbf{v} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (15)$$

$\mathbf{V}^*$  es un vector columna cuyos componentes son las coordenadas transformadas:

$$\mathbf{v} = \begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} \quad (16)$$

Con esta notación, la matriz empleada para la traslación es:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & X_o \\ 0 & 1 & 0 & Y_o \\ 0 & 0 & 1 & Z_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

y el proceso de traslación se completa empleando la ecuación (14), de forma que  $\mathbf{v}^* = \mathbf{T}\mathbf{v}$ .

#### 2.4.1.2. Cambio de Escala

El cambio de escala empleando factores  $S_x$ ,  $S_y$  y  $S_z$  a lo largo de los ejes, X, Y y Z, viene dado por la matriz de transformación:

$$S = \begin{bmatrix} S_x & 0 & 0 & X_o \\ 0 & S_y & 0 & Y_o \\ 0 & 0 & S_z & Z_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

### 2.4.1.3. Rotación

Las transformaciones empleada para la rotación tridimensional son intrínsecamente más complicadas que las transformaciones analizadas hasta ahora. La forma más simple de estas transformaciones es la rotación de un punto sobre un eje de coordenadas. Para girar un punto alrededor de otro punto arbitrario del espacio se requieren tres transformaciones: la primera traslada ese punto arbitrario hasta el origen, la segunda realiza la rotación y la tercera traslada este punto de nuevo a su posición original.

Con respecto al sistema de tres ejes coordenadas, la rotación de un punto un ángulo  $\theta$  alrededor del eje de coordenadas Z se logra empleando la transformación:

$$R_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

El ángulo de rotación  $\theta$  se mide en el sentido de las agujas del reloj cuando se mira al origen desde un punto sobre el eje +Z. Esta transformación afecta únicamente a los valores de las coordenadas X e Y.

La rotación de un punto sobre el eje X con un ángulo  $\alpha$  se realiza empleando la transformación:

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Finalmente, la rotación de un punto, un ángulo  $\beta$ , alrededor del eje Y se consigue mediante la transformación:

$$R_{\beta} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20 \text{ a})$$

## 2.5. La Película Fotográfica

La película fotográfica es un elemento importante de los sistemas de procesamiento de imágenes. A menudo se emplea como medio de registro de las imágenes de entrada, y es con mucho el medio más popular de registrar los productos de salida.

Por estas razones, acabamos este capítulo con una presentación sobre algunas de las propiedades básicas de la película fotográfica monocromática y su relación con las aplicaciones de tratamiento de la imagen.

### 2.5.1. La Estructura de la Película y la Exposición

La sección transversal de película fotográfica típica está formada por las capas y componentes siguientes:

1. Una capa superior de gelatina empleada como protección contra arañazos y abrasiones;
2. Una capa de emulsión con diminutos cristalitas de un haluro de plata;
3. Una capa de sustrato para proporcionar la adhesión de la emulsión a la base de la película;
4. La base, o soporte, de la película, formada por triacetato de celulosa y otro polímero similar, y
5. Un estrato de refuerzo para evitar que se ondule.

### 2.5.2. Características de la Película

De interés práctico para el fotógrafo son los conceptos de contraste, velocidad, tamaño del grano y poder resolutivo. La comprensión de los efectos

de estos parámetros es particularmente importante en aplicaciones especializadas, como fotografiar los resultados obtenidos con un sistema de procesamiento de imágenes.

### **Contraste**

Las películas de alto contraste reproducen las diferencias tonales del sujeto como grandes diferencias de densidad en la fotografía; las películas de bajo contraste convierten las diferencias de tono en pequeñas diferencias de densidad. La exposición,  $E$ , a la que ha sido sometida una película se define como energía por unidad de área en cada punto de la región fotosensible. La exposición depende de la intensidad incidente,  $I$ , y de la duración de la exposición,  $T$ . estas cantidades están relacionadas por la expresión:

$$E = IT$$

### **Velocidad**

La velocidad de una película determina cuanta luz es necesaria para producir una cierta cantidad de plata en el revelado. Cuanto menor es la velocidad, mayor debe ser la exposición para registrar una determinada imagen. El estándar de velocidad mas ampliamente empleado es la escala ASA. Esta escala es aritmética, siendo el valor asignado a la velocidad directamente proporcional a la sensibilidad de la película. Un filme de 200 ASA es el doble de rápido (para un sujeto dado se requiere la mitad de exposición) que una película de 100 ASA. Otras escalas de velocidades, como el sistema DIN empleado en Europa, son logarítmicas. Cada incremento de tres unidades en el número DIN de la velocidad duplica la velocidad. Una película de 50 ASA es equivalente a un DIN 18, un 100 ASA a un DIN 21, y así sucesivamente.

### **Granulosidad**

La imagen extraída de los cristales de haluro de plata tiene una estructura discontinua, lo que le da una apariencia granulosa en grandes ampliaciones. El efecto es más pronunciado en las películas rápidas, que tienen comparativamente cristales mayores; las emulsiones más lentas, de grano más fino, son por tanto preferibles en aplicaciones en las que se necesite un detalle fino o en las que sea necesaria una gran ampliación de los negativos.

### **Poder de Resolución**

La precisión del detalle que puede resolver una película depende no sólo de su granulosis sino también de las propiedades de dispersión de la luz de la emulsión y del contraste con el que la película reproduce los detalles finos. Los mayores poderes de resolución se alcanzan con películas de grano fino y emulsiones delgadas.

### **CAPITULO III RECONOCIMIENTO E INTERPRETACION**

El análisis de imágenes es un proceso que consiste en descubrir, identificar y comprender los patrones que son relevantes en el rendimiento de un trabajo basado en imágenes. Uno de los principales objetivos del análisis de imágenes por computadora consiste en dotar a una maquina, en algún sentido, de la capacidad de aproximación, similar a la de los seres humanos.

Un sistema automático de análisis de imágenes debería ser capaz de ofrecer varios grados de inteligencia. El concepto de inteligencia es algo vago, particularmente con referencia a una maquina. Sin embargo, no resulta difícil expresar el concepto de los diferentes tipos de comportamiento asociados generalmente con la inteligencia. Diversas características nos vienen rápidamente a la mente:

1. La capacidad de extraer la información de interés, separándola de un conjunto de detalles irrelevantes;
2. La capacidad de aprender a partir de ejemplos y de generalizar este conocimiento para que se pueda aplicar en nuevas y diferentes circunstancias, y
3. La capacidad de realizar inferencias a partir de una información incompleta.

Se pueden diseñar e implementar sistemas de análisis de imágenes de estas características para entornos operativos limitados. Sin embargo, no sabemos todavía como dotar a estos sistemas de un nivel de rendimiento que llegue incluso a emular las capacidades humanas a la hora de realizar funciones genéricas del análisis de imágenes.

Gracias a la investigación en sistemas informáticos y biológicos, continuamente se están descubriendo nuevas y prometedoras teorías que intentan explicar el

conocimiento visual humano. Sin embargo, las técnicas más avanzadas de análisis de imágenes por computadora se basan en su mayor parte en fórmulas heurísticas, adaptadas para la resolución de problemas específicos. Por ejemplo, algunas máquinas son capaces de leer documentos impresos, con el formato adecuado, a velocidades superiores en varios órdenes de magnitud a las que pueda conseguir el lector más experto. De cualquier forma, los sistemas de este tipo son sumamente especializados y tiene poca o ninguna aplicación en otras tareas. Esto es, la teoría actual y las limitaciones de implementación en el campo del análisis de imágenes implican soluciones, que, en gran medida, dependen del problema.

### 3.1. Elementos del Analisis de imágenes.

Resulta conceptualmente útil dividir el espectro de técnicas de análisis de imágenes en tres áreas básicas. Estas áreas son:

1. Procesado de bajo nivel,
2. Procesado de nivel intermedio, y
3. Procesado de alto nivel.

Aunque estas subdivisiones no tienen límites definitivos, proporcionan un marco útil para la clasificación de los diversos procesos que son componentes esenciales de un sistema autónomo de análisis de imágenes.

El *procesado de bajo nivel* trata de un tipo de funciones a las que se puede considerar como reacciones automáticas, y que no requieren inteligencia por parte del sistema de análisis de imágenes. Consideramos que la adquisición y el procesado de imágenes son funciones que operan a bajo nivel.

El procesado de nivel intermedio se refiere a la labor de extracción y caracterización de los componentes (por ejemplo, regiones) de la imagen que se obtiene de un proceso a bajo nivel. Es necesario construir algunas capacidades para el comportamiento inteligente dentro de procedimientos flexibles de segmentación. Por ejemplo, para llenar los pequeños huecos de un contorno segmentado es necesario disponer de elementos más sofisticados para la resolución de problema que las simples reacciones automáticas de bajo nivel.

Procesado de alto nivel comprende el reconocimiento y la interpretación. Estos dos procesos tienen un gran parecido con lo que normalmente se conoce como **conocimiento inteligente**. La mayoría de las técnicas empleadas por los

procesados de nivel bajo e intermedio utilizan un conjunto de formulaciones teóricas razonablemente bien definidas. Sin embargo, como se adelantó en el reconocimiento, y especialmente en la interpretación, nuestro conocimiento y comprensión de los principios fundamentales llega a ser mucho menos preciso y mucho más especulativo. Esta relativa falta de comprensión se traduce al final en una formulación de restricciones e idealizaciones cuyo propósito es reducir la complejidad de las tareas, hasta llegar a un nivel razonable. El producto final es un sistema con capacidades operativas sumamente especializadas.

### 3.2. Patrones y Clases de Patrones.

En el análisis de una imagen es fundamental realizar, en algún nivel, un reconocimiento de patrones, Aquí un patrón es una descripción estructural o cuantitativa de un objeto o de alguna otra entidad de interés en una imagen. En general, un patrón está formado por uno o más descriptores. En otras palabras, un patrón es una disposición de descriptores. (En la literatura sobre el reconocimiento de patrones, se suele utilizar la denominación de características para hacer referencia a los descriptores). Una clase de patrones es una familia de patrones que comparten algunas propiedades comunes. Las clases de patrones se representan por  $w_1, w_2, \dots, w_M$ , donde  $M$  es el número de clases. El reconocimiento de patrones mediante una maquina supone la utilización de técnicas que permitan asignar los patrones a sus respectivas clases, automáticamente y con la menor intervención humana posible.

Las tres representaciones de patrones principalmente utilizadas en la práctica son los vectores (para descripciones cuantitativas) y las cadenas y los árboles (para descripciones estructurales). Los patrones vectoriales se representan utilizando letras minúsculas en negrita, como  $x, y, z$ , y tienen la forma:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (21)$$

Donde cada componente,  $x_i$ , representa el  $i$ -ésimo descriptor y  $n$  es el número de descriptores. Los patrones vectoriales se representan como columnas (esto es, como matrices  $n \times 1$ ) De aquí que un patrón vectorial se puede expresar en la forma mostrada por la ecuación (21) o en la forma equivalente  $x = (x_1, x_2, \dots, x_n)^T$ , donde la T indica una transposición.

La naturaleza de los componentes de un patrón vectorial  $x$  depende de la técnica de medida utilizada para describir el propio patrón físico. Por ejemplo, supongamos que queremos describir tres tipos de lirios (*Iris setosa*, *iris virginica* e *Iris versicolor*) por la medida de la longitud y la anchura de sus pétalos. En este caso, utilizaríamos vectores bidimensionales de la forma:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (22)$$

Donde  $x_1$  y  $x_2$  corresponden a la anchura y a la longitud de l pétalo, respectivamente. En este caso, las tres clases de patrones, representadas por  $w_1$ ,  $w_2$  y  $w_3$  corresponden a las variedades *setosa*, *virginica* y *versicolor*, respectivamente.

Puesto que la longitud y la anchura de los pétalos de todas las flores varían en algún grado, los patrones vectoriales que describen estas flores también variarán, no sólo entre clases diferentes, sino también dentro de una misma clase.

### 3.3. Métodos de Decisión Teórica.

Los métodos de decisión teórica para el reconocimiento se basan en la utilización de funciones decisión (o discriminantes). Supongamos que  $x = (x_1, x_2, \dots, x_n)^T$  representa un patrón vectorial de dimensión  $n$ , como se vio anteriormente. Para  $M$  clases de patrones,  $w_1, w_2, \dots, w_m$ , el problema básico en el reconocimiento de patrones por decisión teórica consiste en encontrar  $N$  funciones de decisión  $d_1(x)$ ,  $d_2(x)$ , ...,  $d_M(x)$  que tenga la propiedad de que, si un patrón  $x$  pertenece a la clase  $w_1$ , entonces:

$$d_i(x) < d_j(x) \quad j = 1, 2, \dots, M; \quad j \neq i \quad (23)$$

En otras palabras, un patrón desconocido  $x$  pertenece a la  $i$ -ésima clase de patrones, si al sustituir  $x$  en todas las funciones de decisión,  $d_i(x)$  toma el mayor valor numérico. En caso de empate, los conflictos se resuelven arbitrariamente.

La frontera de decisión que separa la clase  $w_i$  de la  $w_j$  viene dada por valores de  $x$  para los que  $d_i(x) = d_j(x)$  o, expresado de forma equivalente, por los valores de  $x$  para los que:

$$d_i(x) - d_j(x) = 0 \quad (24)$$

La práctica común consiste en hacer que la frontera de decisión entre dos clases sea la función:

$$d_{ij}(x) = d_i(x) - d_j(x) = 0$$

De esta forma,  $d_{ij}(x) < 0$  para los patrones de la clase  $w_i$  y  $d_{ij}(x) > 0$  para los patrones de la clase  $w_j$ .

El principal objetivo de la presentación de esta sección es desarrollar varias técnicas que permitan encontrar funciones de decisión que satisfaga la ecuación (23).

### 3.4. Correspondencias

#### 3.4.1. Clasificador de Mínima Distancia.

Supongamos que cada clase de patrones está representada por un vector prototipo (o medio):

$$m_j = \frac{1}{N_j} \sum_{x \in m_j} x \quad j = 1, 2, \dots, M \quad (25)$$

Donde  $N_j$  es el número de patrones vectoriales de la clase  $w_j$  y la suma se realiza para todos los vectores. Una forma de determinar la pertenencia a una clase de un patrón vectorial desconocido  $x$  consiste en asignarlo a la clase del prototipo más próximo. Si se utiliza la distancia euclídea para determinar el grado de proximidad, se reduce el problema del cálculo de las medidas de distancia:

$$D_j(x) = \|x - m_j\| \quad j=1, 2, \dots, M \quad (26)$$

Donde  $\|a\| = (a^T a)^{1/2}$  es la norma de Euclides. Si  $D_j(x)$  es la menor distancia, entonces se asigna la  $x$  a la clase  $w_j$ . Esto es, en esta formulación, la menor distancia implica el mejor emparejamiento o correspondencia. No resulta difícil demostrar que esto es equivalente a evaluar las funciones.

$$D_j(x) = x^T m_j - \frac{1}{2} m_j^T m_j \quad j = 1, 2, \dots, M \quad (27)$$

Y asignar  $x$  a la clase  $w_j$  si  $d_j(x)$  toma el valor numérico.

Esta formulación concuerda con el concepto de función de decisión, como se ha definido en la ecuación (23).

De las ecuaciones (24) y (27), la frontera de decisión entre las clases  $w_i$  y  $w_j$  para un clasificador de mínima distancia es:

$$\begin{aligned} d_{ij}(x) &= d_i(x) - d_j(x) \\ &= x^T (m_i - m_j) - \frac{1}{2} (m_i - m_j)^T (m_i - m_j) = 0 \quad (28) \end{aligned}$$

La superficie dada por la ecuación (28) es el bisector perpendicular a la línea que une  $m_i$  y  $m_j$ . Para  $n = 2$ , el bisector perpendicular es una línea, para  $n = 3$  es un plano, y para  $n > 3$  un hiperplano.

### 3.4.2. Clasificadores Estadísticos Óptimos. Fundamentos.

En esta sección desarrollaremos una aproximación probabilística al reconocimiento. Como sucede en la mayoría de los campos relacionados con la medición e interpretación física de acontecimientos, las consideraciones probabilísticas tienen su importancia en el reconocimiento de patrones, debido a la aleatoriedad a la que normalmente está sometida la generación de clases de patrones. Como se muestra en la siguiente exposición, es posible obtener una técnica de clasificación óptima en el sentido que, por término medio, su empleo tenga asociada la probabilidad más baja de cometer errores de clasificación.

La probabilidad de que un patrón concreto  $x$ , provenga de una clase  $w_j$  se representa por  $p(w_j/x)$ . Si el clasificador de patrones decide que  $x$  proviene de  $w_j$  cuando realmente proviene de  $w_i$ , incurre en un error, de notación  $L_{ij}$ . Como el patrón  $x$  puede pertenecer a cualquiera de las  $M$  consideradas, el error medio que se produce al asignar  $x$  a la clase  $w_j$  es:

$$r_j(\mathbf{x}) = \sum_{K=1}^M L_{kj} P(w_k/\mathbf{x}) \quad (29)$$

La ecuación (29) se suele denominar riesgo medio condicional o pérdidas según la terminología de la teoría de decisiones.

Según, la teoría básica de probabilidades:

$$p(a/b) = [p(a) p(b/a)] / p(b) \quad (30)$$

Utilizando esta expresión, escribimos la ecuación (30) de la forma:

$$r_j(\mathbf{x}) = \frac{1}{P(\mathbf{x})} \sum_{K=1}^M L_{kj} P(\mathbf{x}/w_k) P(w_k) \quad (31)$$

Donde  $p(\mathbf{x}/w_k)$  es la función densidad de probabilidad de los patrones de la clase  $w_k$  y  $P(w_k)$  es la probabilidad de ocurrencia de la clase  $w_k$ . Puesto que  $1/p(\mathbf{x})$  es positiva y común a todos los  $r_j(\mathbf{x})$ ,  $j = 1, 2, \dots, M$ , puede ser eliminada de la ecuación (31) sin que esto afecte al orden relativo de las funciones, de menor a mayor valor. La expresión del error se reduce entonces a:

$$r_j(\mathbf{x}) = \sum_{K=1}^M L_{kj} P(\mathbf{x}/w_k) P(w_k) \quad (32)$$

El clasificador tiene  $M$  clases posibles para elegir a la hora de clasificar un patrón desconocido dado. Si se calculan  $r_1(\mathbf{x})$ ,  $r_2(\mathbf{x}) \dots r_M(\mathbf{x})$  para cada patrón  $\mathbf{x}$ , y se asigna el patrón a la clase que tenga el menor error, el error medio total con respecto a todas las decisiones será mínimo. El clasificador que minimiza el error medio total se denomina clasificador de Bayes. Así pues, el clasificador de Bayes asigna un patrón  $\mathbf{x}$  a la clase  $w_i$  si  $r_i(\mathbf{x})$  para  $j = 1, 2, \dots, j \neq i$ . En otras palabras, se asigna  $\mathbf{x}$  a la clase  $w_i$  si:

$$\sum_{k=1}^M L_{kj} P(x/w_k) < \sum_{q=1}^M L_{qj} P(x/w_q) P(w_q) \quad (33)$$

En muchos problemas de reconocimiento, el error de una decisión correcta es cero, y el valor no nulo (por ejemplo, 1) para cualquier decisión incorrecta es siempre el mismo. Bajo estas premisas, la función error pasa a ser:

$$L_{ij} = 1 - \delta_{ij} \quad (34)$$

Donde  $\delta_{ij} = 1$  si  $j = i$  y  $\delta_{ij} = 0$  si  $i \neq j$ . La ecuación (34) asigna un error unidad a las decisiones correctas.

Sustituyendo la ecuación (34) en la que (32) se obtiene:

$$\begin{aligned} r_j &= \sum_{k=1}^M (1 - \delta_{kj}) p(x/w_k) P(w_k) \\ &= p(x) - p(x/w_j) P(w_j) \end{aligned} \quad (35)$$

El clasificador de Bayes asigna entonces un patrón  $x$  a la clase  $w_i$ , si:

$$P(x) - p(x/w_i) P(w_i) < p(x) - p(x/w_j) P(w_j) \quad (36)$$

O, de forma equivalente, si.

$$P(x/w_i) P(w_i) > p(x/w_j) P(w_j) \quad j = 1, 2, \dots, M; j \neq i \quad (37)$$

Con referencia a la discusión que conduce a la ecuación (23), vemos que el clasificador de Bayes para funciones de error 0-1 no es nada más que la implementación de las funciones de decisión de la forma:

$$d_j(x) = p(x/w_j) P(w_j) \quad j = 1, 2, \dots, M \quad (38)$$

Donde un patrón vectorial  $x$  se asigna a la clase  $w_i$  si  $d_i(x) < d_j(x)$  para todo  $j \neq i$ .

### 3.5. Redes Neuronales

Las técnicas descritas en las dos secciones anteriores se basan en el uso de patrones ejemplo para la estimación de determinados parámetros estadísticos de cada clase de patrón. El clasificador de mínima distancia queda completamente especificado por el vector medio y la matriz de covarianza de cada clase. A los patrones (de pertenencia conocida a una clase) utilizados para estimar estos parámetros normalmente se los denomina patrones de entrenamiento, y a un conjunto de tales patrones de cada clase se le conoce como conjunto de entrenamiento. El proceso utilizar un conjunto de entrenamiento para obtener funciones de decisión se denomina aprendizaje o entrenamiento.

En los dos métodos descritos, el entrenamiento es algo sencillo. Los patrones de entrenamiento de una determinada clase se utilizan de forma directa para calcular los parámetros de la función de decisión de dicha clase. Después de realizar la estimación de los parámetros en cuestión, se concreta la estructura del clasificador, y su rendimiento dependerá de cómo se ajustan las distribuciones de los patrones reales a las suposiciones estadísticas realizadas en la obtención del método de clasificación utilizado.

A menudo, las propiedades estadísticas de las clases de patrones de un problema son desconocidas, o no es posible realizar una estimación de las mismas (recuérdense lo que se dijo en la sección anterior sobre la dificultad de trabajar con estadística multivariable). En la práctica, estos problemas de decisión se gestionan mejor utilizando métodos que obtienen directamente las funciones de decisión requeridas mediante el entrenamiento. Así pues, no es necesario realizar suposiciones sobre las funciones de densidad de probabilidad subyacentes, o sobre cualquier otra información estadística relativa a las clases de patrones consideradas. En esta sección estudiaremos varias técnicas que siguen este criterio.

La esencia del material que viene a continuación es la utilización de una multitud de elementos de cálculo no lineales y elementales (denominados neuronas), organizados como redes que se asemejan a la forma en que se

piensa que están interconectadas a las neuronas en el cerebro. Los modelos resultantes poseen diversas denominaciones, como pueden ser redes neuronales, neurocomputadoras, modelos de procesamiento paralelo distribuido (PDP, del inglés: parallel distributed processing), sistemas neuromórficos, redes multicapa autoconfigurables y modelos conexionistas. Aquí utilizaremos la denominación "redes neuronales". Usaremos estas redes como vehículos para el desarrollo de los coeficientes de las funciones de decisión mediante la presentación sucesiva de conjuntos de patrones de entrenamiento.

El interés de las redes neuronales data de principios de los años cuarenta, como queda de manifiesto en el trabajo de McCulloch y Pitts (1943). Estos autores propusieron modelos neuronales en forma de dispositivos binarios de umbralización y algoritmos estocásticos que implican transiciones repentinas 0-1 y 1-0 en el estado de las neuronas, como base para el modelado de sistemas neuronales. Posteriormente, el trabajo de Hebb (1949) se basó en modelos matemáticos que intentaron abordar el concepto del aprendizaje mediante el refuerzo y la asociación.

Durante la mitad de los años cincuenta y a principios de los sesenta, un tipo de sistemas denominados máquinas de aprendizaje, ideados por Rosenblatt (1957, 1962), crearon grandes expectativas entre los investigadores y quienes ponían en práctica la teoría de reconocimiento de patrones. El motivo del gran interés por estas máquinas, denominadas perceptrones, fue el desarrollo de demostraciones matemáticas que probaron que al entrenar a los perceptrones utilizando conjuntos de entrenamiento linealmente separables, éstos convergen a una solución en un número finito de iteraciones. Las soluciones tomaban la forma de coeficientes de hiperplanos capaces de separar correctamente las clases representadas por los patrones del conjunto de entrenamiento.

Por desgracia, las esperanzas creadas con el descubrimiento de lo que parecía ser un buen modelo teórico del aprendizaje pronto se desvanecieron. El perceptrón básico y algunas de sus generalizaciones de aquel entonces eran simplemente insuficientes para la mayoría de las tareas de reconocimiento de patrones de significación práctica. Los posteriores intentos de mejora de la potencia de máquinas de tipos perceptrón considerando múltiples niveles de estos dispositivos, aunque eran conceptualmente atractivos, carecían de

algoritmos de entrenamiento efectivos, como los que habían creado el interés en el propio perceptrón. Nilsson (1965) resumió el estado de las investigaciones en el campo de las máquinas de aprendizaje a mediados de los años sesenta. Unos pocos años después, Minsky y Papert (1969) presentaron un análisis descorazonador sobre la limitación de las máquinas de tipo perceptrón. Este punto de vista imperó hasta mitad de los ochenta, como se desprende de los comentarios de Simon (1986). En esta obra, originalmente publicada en francés en 1984, Simon descarta el perceptrón utilizando el título "Nacimiento y muerte de un mito".

Los recientes resultados obtenidos por Rumelhart, Hinton y Williams (1986) en el desarrollo de nuevos algoritmos de entrenamiento para perceptrones multicapa han cambiado las cosas de forma considerable. El método básico que emplearon a menudo conocido como regla delta generalizada para el aprendizaje por retropropagación, ofrece una técnica de entrenamiento efectiva para máquinas multicapa. Aunque no es posible demostrar que este algoritmo análogo para el perceptrón de un nivel, se ha podido utilizar con éxito la regla delta generalizada en varios problemas de interés práctico. Con este éxito se ha conseguido que las máquinas tipo perceptrón multicapa constituyen uno de los principales modelos de redes neuronales actualmente en uso.

Aunque las investigaciones sobre nuevas reglas de entrenamiento para máquinas multicapa, los resultados de nuevos modelos de redes neuronales y algunas aplicaciones de resultados han encendido la chispa de un renovado interés por el reconocimiento de patrones basado en máquinas de aprendizaje, la investigación en este campo está aún poca avanzada. El cerebro humano posee en torno a 100 billones de neuronas, organizadas mediante una red compleja en la que las neuronas individuales pueden estar conectadas a varios miles de neuronas distintas. Aún no se sabe cómo esta red masivamente paralela, en la que las neuronas individuales se disparan a velocidades medidas en milisegundos (en comparación con las velocidades del orden de nanosegundos de los componentes electrónicos), puede realizar con asombrosa rapidez tareas como la adquisición, almacenamiento, representación, recuperación y análisis de datos sensoriales sumamente complejos. Es todo un ejercicio de humildad pensar en esta exquisita capacidad del cerebro para el tratamiento de datos, y la

forma en que nos permite construir imágenes mentales de personas, lugares o sucesos basados en fragmentos de información, tales como un sonido o un dolor familiar, o simplemente el recuerdo de una fecha.

En este contexto, lo que se ha conseguido hasta ahora en el campo de las redes neuronales es, en comparación, ridículo. Por tanto, los retos que quedan por delante este campo de investigación son muchos. Aquí, nuestra tarea consiste en introducir diversos aspectos del estado de la tecnología de la implementación de redes neuronales multicapa. Empezaremos estudiando el perceptrón como modelo básico de una neurona. Posteriormente trataremos varios aspectos del entrenamiento de perceptrones en los supuestos de separabilidad de las clases y de no separabilidad de las mismas. Este material sirve como base para el desarrollo e ilustración de la regla delta generalizada para el entrenamiento de redes neuronales multicapa.

### 3.6. Métodos Estructurales

Las técnicas expuestas anteriormente se ocupan de los patrones de forma cuantitativa, e ignoran totalmente cualquier relación estructural inherente a las formas de los mismos. Sin embargo, los métodos estructurales que estudiaremos en esta sección tratan de lograr el reconocimiento del patrón poniendo el énfasis precisamente en este tipo de relaciones.

#### 3.6.1. Correspondencia de Números de Forma

Se puede formular un procedimiento análogo al concepto de la mínima distancia, visto anteriormente, aplicable a patrones vectoriales, para la comparación de contornos que se describen mediante números de forma. Con referencia a lo que se estudió anteriormente, el grado de similitud,  $k$ , entre dos contornos (formas),  $A$  y  $B$ , se define como el mayor orden para el que aun coinciden sus correspondientes números de forma. Por ejemplo, en el caso de los números de forma de los límites cerrados representados por los códigos de cadena de 4 direcciones,  $A$  y  $B$  poseen un grado  $k$  de similitud si  $S_4(A) = S_4(B)$ ,  $S_5(A) = S_5(B)$ ,  $S_6(A) = S_6(B)$ ,  $S_7(A) = S_7(B)$ , ...,  $S_k(A) = S_k(B)$ ,  $S_{k+2}(A) \neq S_{k+2}(B)$ ,  $S_{k+4}(A) \neq S_{k+4}(B)$ , ..., donde  $S$  indica el número de forma y el subíndice indica el orden. Se define la distancia entre dos formas  $A$  y  $B$  como lo inverso de su grado de similitud.

$$D(A,B) = \frac{1}{K} \quad (39)$$

Esta distancia tiene las siguientes propiedades:

$$\begin{aligned} D(A,B) &\geq 0 \\ D(A,B) &= 0 \text{ si } A = B \\ D(A,C) &\leq \text{máx} [D(A,B), D(B,C)] \end{aligned} \quad (40)$$

Se pueden utilizar tanto  $k$  como  $D$  para comparar dos formas. Si se utiliza el grado de similitud, cuando mayor sea  $k$ , más similares son las formas (obsérvense que  $k$  es infinito para formas idénticas). Lo inverso es cierto cuando se utiliza la medida de la distancia.

### 3.6.2 Correspondencia de Cadenas

Supóngase que se codifican dos contornos  $A$  y  $B$  en las cadenas representadas por  $a_1a_2\dots a_n$  y  $b_1b_2\dots b_n$ , respectivamente. Supóngase también que  $M$  representa el número de correspondencias entre ambas cadenas, teniendo en cuenta que se produce una correspondencia en la  $k$ -ésima posición si  $a_k = b_k$ . El número de símbolos que nos e corresponden es:

$$Q = \text{máx} (|A|, |B|) - M \quad (41)$$

Donde  $|arg|$  es la longitud (número de símbolos) de la representación en forma de cadena del argumento. Se puede demostrar que  $Q = 0$  si, y sólo si,  $A$  y  $B$  son idénticos.

Una sencilla medida de la similitud entre  $A$  y  $B$  es el cociente.

$$R = \frac{M}{Q} = \frac{M}{\text{máx} (|A|, |B|) - M} \quad (42)$$

Por lo tanto,  $R$  es infinito si existe una correspondencia total, y vale 0 cuando no hay correspondencias entre los símbolos de  $A$  y  $B$  (en este caso,  $M=0$ ). Puesto que la comparación se hace símbolo, el punto inicial de cada contorno es importante para la reducción del número de cálculos. Cualquier método capaz de realizar una normalización al mismo punto, o a puntos próximos, será de gran utilidad, al ofrecer una ventaja de cálculo frente al método de correspondencia por fuerza bruta, que consiste en comenzar en puntos arbitrarios de cada cadena, ir desplazando cíclicamente una de las cadenas y calcular la ecuación (42) para cada desplazamiento. El mayor grado de correspondencia se obtiene con el mayor valor de  $R$ .

### **3.7. Métodos Sintácticos**

Los métodos sintácticos se encuentran entre las técnicas más utilizadas en la gestión de problemas de reconocimiento de estructuras. Básicamente la idea del reconocimiento sintáctico de patrones consiste en la especificación de un conjunto de primitivas de patrones, un conjunto de reglas (en forma de gramática) que gobierna su interrelación, y un reconocedor (denominado autómatas) cuya estructura está determinada por el conjunto de reglas que impone la gramática. En primer lugar, estudiaremos las gramáticas y los autómatas de cadenas, y posteriormente extenderemos estos conceptos a las gramáticas de árbol y sus correspondientes autómatas. Se recuerda que en este trabajo las cadenas y los árboles son los principales descriptores estructurales de los patrones.

### **3.8. Interpretación de Imágenes**

Hasta ahora hemos mencionado principalmente de procesos individuales, que varían desde la adquisición y el preprocesado de imágenes hasta la segmentación, la descripción y el reconocimiento de objetos. En esta sección abordaremos el problema de la utilización de la información generada por todos estos procesos para intentar interpretar el contenido de una imagen. En otras palabras, estamos interesados en dar un significado a una imagen, un proceso que, además de interpretación, también suele denominarse comprensión de imágenes o análisis de escenas. Por coherencia con las explicaciones anteriores, continuaremos utilizando la denominación interpretación de imágenes.

### 3.8.1. Fundamentos

La interpretación de imágenes por medio de computadora es un proceso sumamente complejo. Las dificultades se producen tanto por la gran cantidad de datos que se deben procesar como por la falta de herramientas básicas de procesamiento necesarias para pasar desde lo que se recibe (una matriz de pixels) hasta el resultado deseado (una lista detallada del contenido de la imagen). Por tanto, ya que no existen herramientas generales que hagan posible la interpretación de imágenes no estructuradas, nos vemos forzados a considerar técnicas que ofrezcan una razonable posibilidad de éxito. Esta restricción nos conduce a un compromiso de dos etapas:

1. Limitar la generalidad del problema, y
2. Intentar incorporar el conocimiento del ámbito del problema es, en principio, sencilla.

Cuando se pueda, utilizaremos todos los medios disponibles para limitar las circunstancias desconocidas con el fin de simplificar el problema que nos ocupa. Cuando no se pueda ejercer un control sobre las variables del problema, nos veremos forzados a limitar el abanico(y la precisión) de resultados esperados.

Los métodos semánticos representan el conocimiento en forma de lo que se conoce como redes semánticas que son grafos dirigidos, con leyendas. Estos grafos ofrecen formulaciones intuitivas en términos de relaciones entre los elementos de una imagen. En este tipo de formulación, los objetos se representan como nodos de un grafo, y las relaciones entre los mismos se representan como arcos que conectan los distintos nodos y que poseen una leyenda. Como veremos mas adelante, las redes semánticas son una potente herramienta para la interpretación de imágenes

Los métodos basados en los sistemas de producción (también conocidos sistemas basados en reglas), han recibido la mayor atención en las aplicaciones de interpretación de imágenes. Los motivos de esta atención son la disponibilidad de numerosas herramientas para el desarrollo de estos sistemas, y el hecho de que en ellos se puede aplicar el conocimiento humano de forma intuitiva, directa y gradual. En concreto, los sistemas expertos, que están restringidos a tareas específicas, son capaces de abordar un amplio abanico de aplicaciones de imágenes.

### 3.8.2. Tipos de Conocimiento

La subdivisión de las funciones del procesado de imágenes en tres categorías (proceso de bajo, intermedio y alto nivel) descrita anteriormente es válida para los algoritmos, pero no resulta particularmente útil para clasificar los grados de conocimiento necesarios para realizar una tarea de procesado de imágenes. Una clasificación bastante más útil puede ser:

1. Conocimiento procedimental,
2. Conocimiento visual, y
3. Conocimiento universal.

**El conocimiento procedimental** se refiere a operaciones tales como la selección de algoritmos y el ajuste de los parámetros de los mismos (por ejemplo, la selección de los valores de umbral). **El conocimiento visual** trata de los distintos aspectos de la formación de imágenes, como puede ser el concepto de que al iluminar oblicuamente un objeto tridimensional se debe esperar la aparición de una sombra. **El conocimiento universal** indica el conocimiento global en el ámbito de un problema. Ejemplos de este tipo de conocimiento pueden ser las relaciones conocidas entre los objetos de una imagen (por ejemplo, en la imagen aérea de un aeropuerto, las pistas de aterrizaje se cruzan con las de rodaje) y las relaciones entre una escena y su entorno (como el hecho de que, de noche, la lluvia incrementa la reflectividad de las carreteras).

### 3.8.3. Sistemas Lógicos (Cálculo de predicados)

La lógica de predicados ha evolucionado durante más de un siglo, hasta la consecución de un extenso y bien documentado conjunto de conocimientos útiles para la expresión de proposiciones y para la inferencia de nuevos hechos a partir de una base de conocimiento. Uno de los elementos más potentes de este conjunto de conocimientos es el cálculo de predicados de primer orden, un tipo de lógica capaz de gestionar un amplio rango de expresiones matemáticas, así como sentencias en un lenguaje natural, como el español.

### 3.8.4. Redes Semánticas

Las redes semánticas son grafos dirigidos, y etiquetados, en los que los nodos normalmente representan objetos y variables y los arcos representan relaciones entre dichos nodos. Las redes de este tipo tienen varias ventajas, entre las que se encuentran la representación intuitiva y visualmente efectiva del

conocimiento. Además, puesto que la representación básica es un grafo, se pueden utilizar técnicas de concordancia y etiquetado de grafos para tratar los elementos del dominio de un problema.

Las redes semánticas ofrecen una forma alternativa de realizar tareas de interpretación. Aunque a tal efecto es posible utilizar las redes de muchas formas, en esta sección nos centraremos en las redes que se emplean para la representación y el procesado del conocimiento expresado en forma clausulada. Este enfoque ilustra los conceptos básicos subyacentes en la utilización de la presentación.

La construcción de una red semántica a partir de expresiones en forma clausulada es inmediata. La idea consiste en representar cada cláusula como una partición de la red semántica (es decir, cada cláusula se representa como un subgrafo independiente). Los nodos de la red contienen variables y constantes (normalmente, objetos), y los arcos orientados y etiquetados representan relaciones binarias entre dichos nodos. Un arco se dibuja como una flecha de una sola punta si representa una relación en la parte de la conclusión (derecha) de una cláusula, y como una flecha con dos puntas si representa una relación en la parte de la condición (izquierda) de una cláusula.

### **3.8.5. Sistemas Expertos de Producción**

Los sistemas de producción ofrecen un enfoque alternativo a los tipos de inferencia analizados anteriormente. Como el cálculo de predicados y las redes semánticas, los sistemas de producción requieren una concordancia para poder identificar la inferencia a realizar. Sin embargo, las acciones de un sistema de producción después de haber encontrado una concordancia son mucho más generales. De hecho, se permiten acciones de complejidad arbitraria.

Esta flexibilidad de respuesta ha provocado un gran interés en el desarrollo de sistemas de producción. En particular, los sistemas expertos han sido utilizados con éxito en diversas aplicaciones, desde el procesado de imágenes y la inspección industrial hasta los diagnósticos médicos y el control de procesos. Los sistemas expertos son sistemas que poseen una cierta pericia (especializada) para la resolución de problemas. La denominación sistemas expertos refleja el hecho de que, generalmente, estos sistemas se basan en el conocimiento obtenido por personas expertas en una disciplina específica.

## **CAPITULO IV APLICACIONES EN MATLAB UTILIZANDO TOOLBOX DE PROCESAMIENTO DE IMAGENES**

### **4.1. Introducción al Análisis y Mejoramiento de Imágenes**

La caja de herramientas (Toolbox) soporta o apoya un rango normal de operaciones de procesamiento de imágenes para analizar y mejorar las imágenes. Sus funciones simplifican varias categorías de tareas entre otras:

1. Obtener el valor del Pixel y valores estadísticos, que son resúmenes numéricos de los datos de una imagen.
2. Analizar imágenes para extraer información acerca de su estructura esencial
3. Mejorar las imágenes haciendo más fácil de ver ciertos rasgos de la imagen o reducir el ruido de la misma.

#### **4.1.1. Los Valores del Pixel y Estadísticas**

La Caja de herramientas de imágenes proporciona varias funciones que retornan la información sobre los valores de los datos que constituyen una imagen. Entre estas funciones se tiene:

1. Devuelva la información sobre los datos de la imagen en varios formularios, incluyendo los valores de los datos para los pixels seleccionados (el `pixval`, `impixel`)
2. Los valores de los datos de un camino en una imagen (el `improfile`)
3. Dibujar el contorno de la imagen (`imcontour`)
4. El histograma de la imagen (`imhist`)
5. Un resumen estadístico de una imagen (el `mean2`, el `std2`, el `corr2`)
6. Medir las características o rasgos de una región de la imagen (`imfeature`)

#### 4.1.2. Selección del pixel

La caja de herramientas incluye dos funciones que proporcionan la información sobre el color de los pixels de la imagen que usted especifica:

**pixval** displaya interactivamente los valores de los datos para los pixeles cuando usted mueve el cursor encima de la imagen. los pixval también pueden visualizar la distancia euclidiana entre dos pixeles.

**impixel** retorna el valor de los datos para un pixels o colección de pixeles. Ud. puede suministrar las coordenadas de los pixeles como argumentos de entrada o Ud. puede seleccionar los pixeles usando un mouse.

En este ejemplo, Ud. invoca **impixel** y da click a tres puntos de la imagen displayada y luego presiona Return

```
Imshow canoe.tif
```

```
Vals=impixel
```

canoe.tif



canoe.tif



```
Vals =  
0.1922  0.2235  0.1922  
0.3882  0        0  
0.2902  0.2588  0.2235
```

notar que el segundo pixel, que es parte de la canoa, es rojo, el verde y azul son ceros para imágenes indexadas `pixval` y `impixel` muestran los valores RGB almacenados en el `COLORMAP`, no en el valor indexado

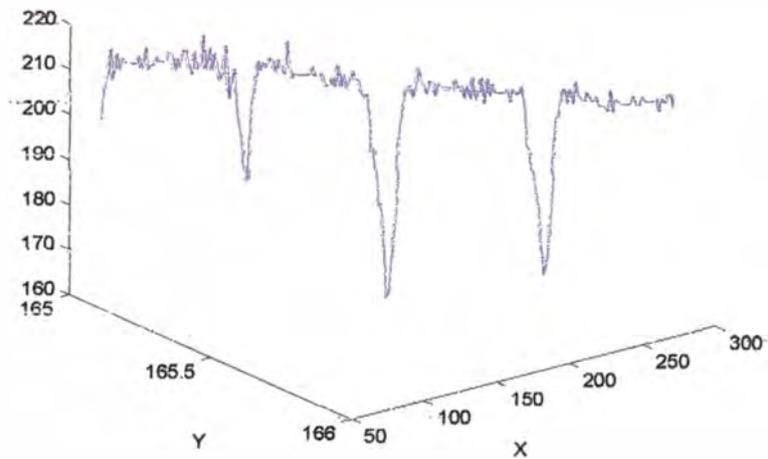
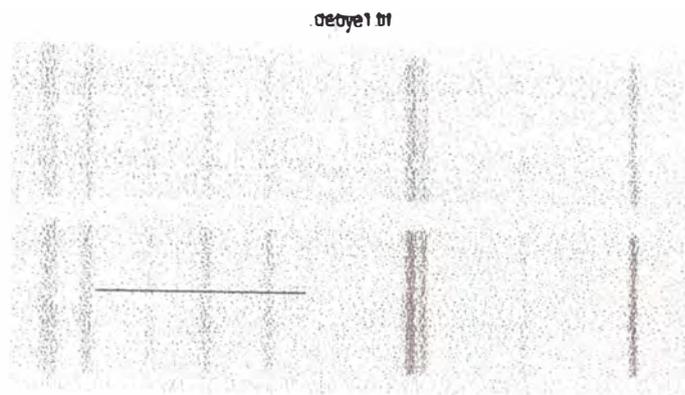
#### 4.1.3. El Perfil de Intensidad

La función `improfile` calcula y traza o dibuja los valores de intensidad a lo largo de un segmento de línea o un camino de múltiples líneas en una imagen. Usted puede proporcionar las coordenadas de los segmentos de la línea como argumentos de entrada, o usted puede definir el camino deseado usando un mouse

En este ejemplo Ud. invoca `improfile` y especifica una simple línea con el mouse

```
Imshow debye1.tif
```

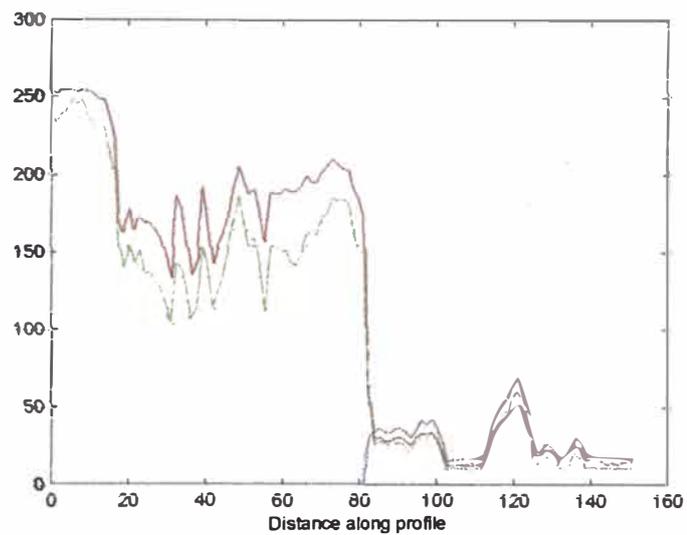
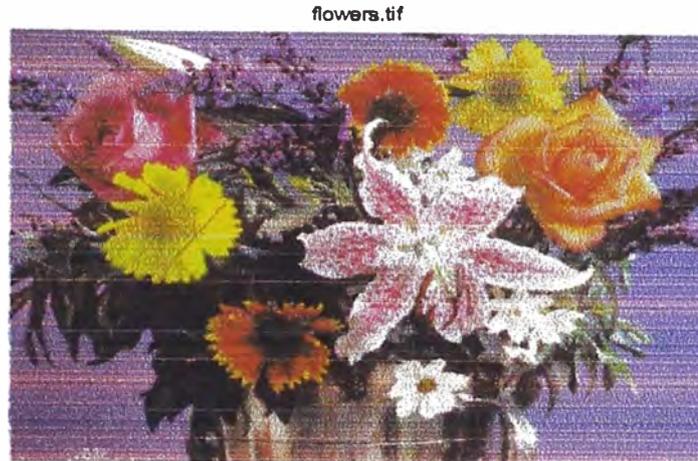
```
Improfile
```



Ejemplo Se muestra como trabaja improfile con una imagen RGB. La línea roja es dibujada de arriba abajo.

```
imshow flowers.tif
```

```
improfile
```



#### 4.1.4. Contornos de las imágenes

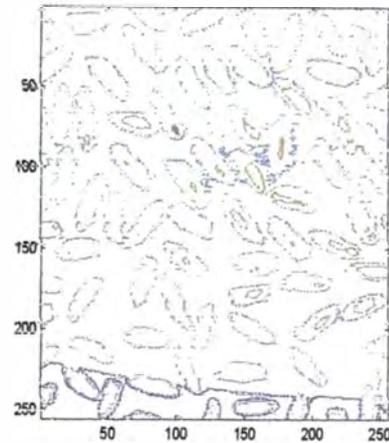
Usted puede usar el `imcontour` de caja de herramientas para visualizar el contorno de la intensidad de una imagen de intensidad. Esta función es similar a la función del contorno en MATLAB.

Este ejemplo despliega una imagen en la escala de grises de los granos de arroz y una porción del contorno de los datos de la imagen.

```
I = imread ('rice.tif');
```

```
imshow(I)
```

```
figure, imcontour(I)
```



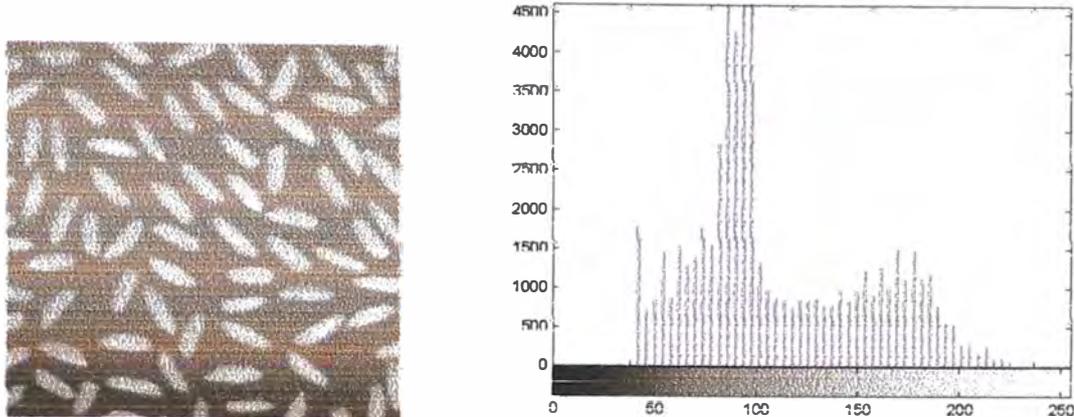
Usted puede usar que los **clabel** funcionan para etiquetar los niveles de los contornos. Vea el la descripción de **clabel** en la MATLAB.

#### 4.1.5. El Histograma de la Imagen

Un histograma de una imagen es un diagrama, en que muestra la distribución de intensidades en una imagen de índice o de intensidad. La función imagen histograma **imhist** crea esto trace haciendo  $n$  cajas de espacios iguales, cada uno que representa un rango de valores de los datos.

Calcula el número de pixeles entonces dentro de cada rango. Por ejemplo, el siguiente comando despliega una imagen de granos de arroz, y un histograma basado en 64 cajas

```
l = imread ('rice.tif');
imshow(l)
figure, imhist(l,64)
```



El histograma muestra una cresta a alrededor de 100, debido al fondo gris oscuro en la imagen.

Para la información sobre cómo modificar una imagen cambiando la distribución de su histograma, vea el Ajuste de Intensidad

## 4.2. Las Estadísticas Sumarias

Usted puede computar estadísticas de la imagen normales que usan el `mean2`, el `std2`, and `corr2`, las funciones `mean2` y `std2` computan la media, la desviación standard de los elementos de una matriz `corr2` computa el coeficiente de la correlación entre dos matrices del mismo tamaño.

Estas funciones son las versiones bidimensionales de la media, la desviación estándar y el coeficiente de correlación.

### 4.2.1. Características de Medida

La función `imfeature` permite computar las medidas del rasgo para regiones de una imagen. Por ejemplo, los `imfeature` pueden medir cosas el área, el centro de masa y limitando la caja para una región usted especifique.

### 4.2.2. El Análisis de la Imagen

Las técnicas de análisis de imagen devuelven la información sobre la estructura de una imagen. Esta sección describe funciones de la caja de herramientas que usted puede usar:

El descubrimiento o detección del borde

#### 1. La descomposición de **Quadtree**

Las funciones sólo trabajan con imágenes de intensidad de escala de grises

## El Descubrimiento o Detección del Borde

Usted puede usar la función del borde para descubrir bordes en que son esos lugares de una imagen que corresponde a las fronteras o límites del objeto. Para encontrar los bordes, estas funciones buscan lugares en la imagen dónde la intensidad cambia rápidamente, usando uno de éstos dos criterios:

- Lugar dónde la primera derivada de la intensidad es más grande en magnitud que algún umbral.
- Lugar dónde la segunda derivada de la intensidad tiene un cruce en cero

El borde proporciona varios estimadores derivativos cada uno de los cuales lleva a cabo una de las definiciones anteriores. Para algunos de estos estimadores, usted puede especificar si el funcionamiento debe ser sensible a los bordes horizontales o verticales, o ambos. El borde retorna una imagen binaria que contiene 1's donde se encuentran los bordes y 0's en otra parte.

El método de borde-descubrimiento más poderoso es el método Canny. El método Canny difiere de los otros métodos de borde-descubrimiento en que usa dos umbrales diferentes (para detectar bordes bien trazados y débiles), y sólo incluye los bordes débiles en la salida si ellos se conectan muy bien a los bordes bien trazados.

El ejemplo debajo de ilustra la potencia del método Canny. Muestra los resultados de aplicar los Sobel y Canny a la imagen rice.tif

```
I = imread ('rice.tif');
BW1 = edge(I,'sobel');
BW2 = edge(I,'canny');
imshow(BW1)
figure, imshow(BW2)
```



### Descomposición de Quadtree

Es una técnica del análisis que involucra subdividir una imagen en bloques que son más homogéneos que la propia imagen. Esta técnica revela la información sobre la estructura de la imagen. También es útil como el primer paso en los algoritmos adaptivos de compresión.

Usted puede realizar descomposición del **quadtree** que usa la función `qtdecomp`. Esta función trabaja dividiendo una imagen cuadrada en cuatro bloques cuadrados de igual tamaño y luego prueba cada bloque para ver si se encuentra algún criterio de homogeneidad (por ejemplo, si todos los píxeles en el bloque están dentro de un rango dinámico específico). Si en un bloque se encuentra el criterio, éste no es dividido más. Si no se encuentra el criterio, se subdivide de nuevo en cuatro bloques, y el criterio de la prueba es aplicado a esos bloques. Este proceso se repite iterativamente hasta que cada bloque encuentre el criterio. El resultado puede tener bloques de varios tamaños diferentes.

Por ejemplo, suponga que se aplica el método a una imagen de escala de gris de 128 por 128. La primera etapa es dividir la imagen en cuatro bloques de

64 por 64. Luego aplica la prueba de criterio para cada bloque, por ejemplo el criterio podría ser:

$$\text{Max}(\text{block}(:)) - \text{min}(\text{block}(:)) \leq 0.2$$

Si uno de los bloques reúne este criterio, éste no es dividido más, los bloques que no cumplen con el criterio se subdividen (algunos de los bloques son tan pequeños como 1 por 1)

La invocación a la función podría ser:

$$S = \text{qtdecomp}(I, 0.2)$$

S es retornado como una matriz sparse cuyos elementos no ceros representan parte superior izquierdo de el bloque, S es del mismo tamaño que I

### 4.3. Mejorando la Imagen

Se usan las técnicas de mejora de imagen para mejorar una imagen dónde "mejore" a veces se define objetivamente (por ejemplo, aumente la razón señal-a-ruido), y a veces subjetivamente (por ejemplo, haga ciertos rasgos más fácil para ver modificando los colores o intensidades).

Esta sección discute éstos las técnicas de mejora de imagen:

1. El ajuste de Intensidad
2. Eliminando el ruido

Las funciones descritas en esta sección aplican principalmente a la intensidad de imágenes en la escala de gris. Sin embargo, algunas de estas funciones pueden aplicarse para imágenes de color

#### 4.3.1. El Ajuste de Intensidad

El ajuste de intensidad es una técnica para mapear la intensidad de una imagen valorada a un nuevo rango. Por ejemplo, mire la imagen del arroz y su histograma en el ejemplo anterior.

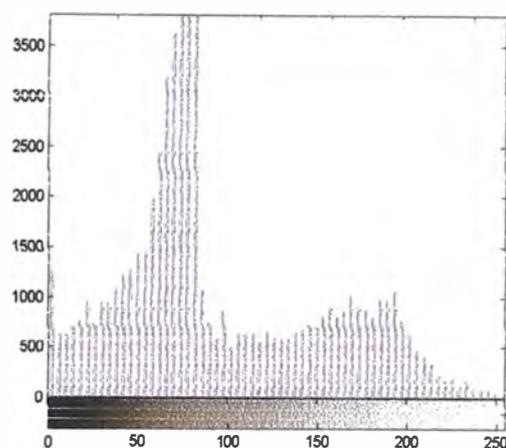
Se nota que la imagen tiene el contraste bastante bajo, y que el histograma indica ningún valor debajo de 40 o arriba de 255. Si usted el **remapes** los datos llenando el rango entero de intensidad [0,255], usted puede aumentar el contraste de la imagen. Usted puede hacer este tipo de ajuste con la función **imadjust**. Por ejemplo, éste código realiza el ajuste descrito anteriormente:

```
I = imread('rice.tif');
J = imadjust(I,[0.15 0.9],[0 1]);
```

Notar que las intensidades son especificadas como valores entre 0 y 1 , indiferente, de la clase de I. Si I es uint8, los valores que son suministrados son multiplicados por 255 para determinar el valor actual a usar; si I es uint16, los valores son multiplicados por 65535. Ahora visualice la imagen ajustada y su histograma:

```
imshow(J)
```

```
figure, imhist(J,64)
```



Se nota el incremento de contraste en la imagen, y que el histograma cubre el rango entero. Similarmente, se puede disminuir el contraste de una imagen reduciendo el rango de la data:

```
J=imadjust(I,[0 1],[0.3 0.8])
```

La sintaxis general es:

```
J=imadjust(I,[low high],[bottom top])
```

Incrementando y disminuyendo el contraste puede mejorar la imagen. Por ejemplo el saco del hombre está demasiado oscuro para revelar ciertos detalles. Invocando la función `imadjust` el rango `[0,51]` en la imagen de entrada uint8, a `[128,255]` en la imagen de salida. La brillantez de la imagen es considerable haciendo más fácil de ver los detalles del saco.

```
I=imread('cameraman.tif');
```

```
J=imadjust(I,[0 0.2],[0.5, 1]);
```

```
imshow(I)
```

figure, imshow(J)



#### 4.3.2. Corrección Gamma

Los `imadjust` pueden aceptar un argumento adicional que especifica el factor de la corrección gamma. Dependiendo del valor de gamma, el mapeo de los valores de la imagen de entrada y salida puede ser no lineal. Gamma puede tener valores entre cero y infinito. Si gamma es 1 - por defecto- el mapeo es lineal. Si gamma es menor que 1, el mapeo tiene más peso hacia los valores de salida alto (brillante). Si gamma es mayor que 1, el mapeo tiene más peso a los valores de salida bajo (oscuro)

Los ejemplos siguientes ilustran la corrección gamma. Notar que cuando se invoca la función `imadjust`, los rangos de datos de las imágenes de entrada y salida son especificadas como matrices vacías. Cuando se especifica una matriz vacía, `imadjust` utiliza por defecto el rango [0 1]:

```
[X,map]=imread('forest.tif')
I=ind2gray(X,map);
J=imadjust(I,[],[],0.5);
imshow(I)
figure, imshow(J)
```



### **Ecuación del Histograma**

El proceso de ajustar los valores de intensidad puede hacerse automáticamente con la función `histeq` ejecuta el histograma de ecualización, que involucra transformar el valor de intensidad para que el histograma de la imagen de salida se iguale a un histograma específico (Por defecto, el `histeq` intenta a empareje a un histograma plano con 64 cajas, pero usted puede especificar un histograma diferente).

Este ejemplo ilustra el uso del `histeq` para ajustar una imagen de intensidad. La imagen original tiene contraste bajo, con la mayoría de los valores en la mitad del rango de la intensidad. el `histeq` produce una imagen de salida que tiene los valores uniformemente distribuido a lo largo del rango.

```
I = imread('pout.tif');
```

```
J = histeq(I);
```

```
imshow(I)
```

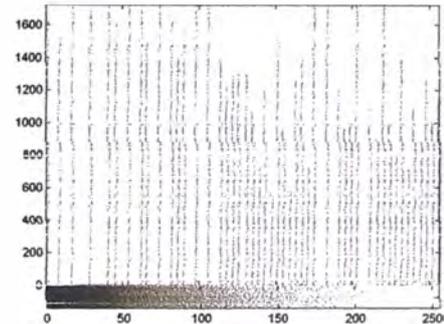
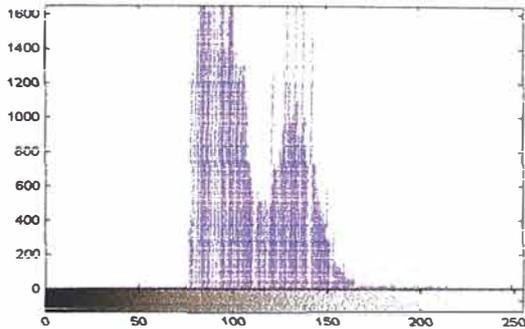
```
figure, imshow(j)
```



A continuación se muestran el histograma para las dos imágenes:

```
Imhist(I)
```

```
figure, imhist(J)
```

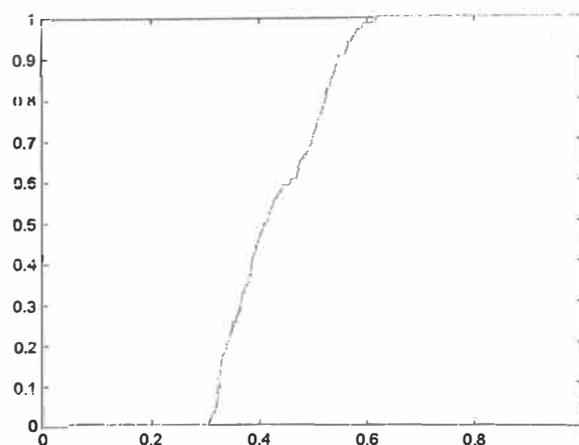


El `histeq` puede retornar un adicional vector 1 por 256 que para cada valor posible de la entrada el valor de la salida (los valores en este vector están en el rango[0.1], sin tener en cuenta la clase de la imagen de la entrada.) Usted puede dibujar estos datos para conseguir la curva de transformación:

```
I = imread('pout.tif');
```

```
[J,T] = histeq(I);
```

```
plot((0:255)/255,T);
```



Notar que esta curva refleja el histograma en la figura anterior, con los valores de entrada principalmente entre 0.3 y 0.6, mientras los valores de salida son distribuidos uniformemente entre 0 y 1.

#### **4.4. La Eliminación del Ruido**

Las imágenes digitales son propensas a una variedad de tipos de ruido. Hay varias maneras en que los ruidos puede introducirse en una imagen, dependiendo de cómo la imagen es creada. Por ejemplo:

- Si la imagen es escaneada de una fotografía hecha en la película, el grano cinematográfico es un la fuente de ruido. El ruido también puede ser el resultado de daño a la película, o es introducido por el propio escáner.
- Si la imagen es directamente adquirida en un formato digital, el mecanismo para recoger los datos puede introducir el ruido.

La transmisión electrónica de datos de la imagen puede introducir el ruido.

La caja de herramientas proporciona varios maneras diferentes para eliminar el ruido en una imagen. Los métodos diferentes son buenos para los tipos diferentes de ruido. Los métodos disponibles son:

- Filtrado Lineal
- Filtrado Mediano
- Filtrado Adaptivo

##### **Filtrado Lineal**

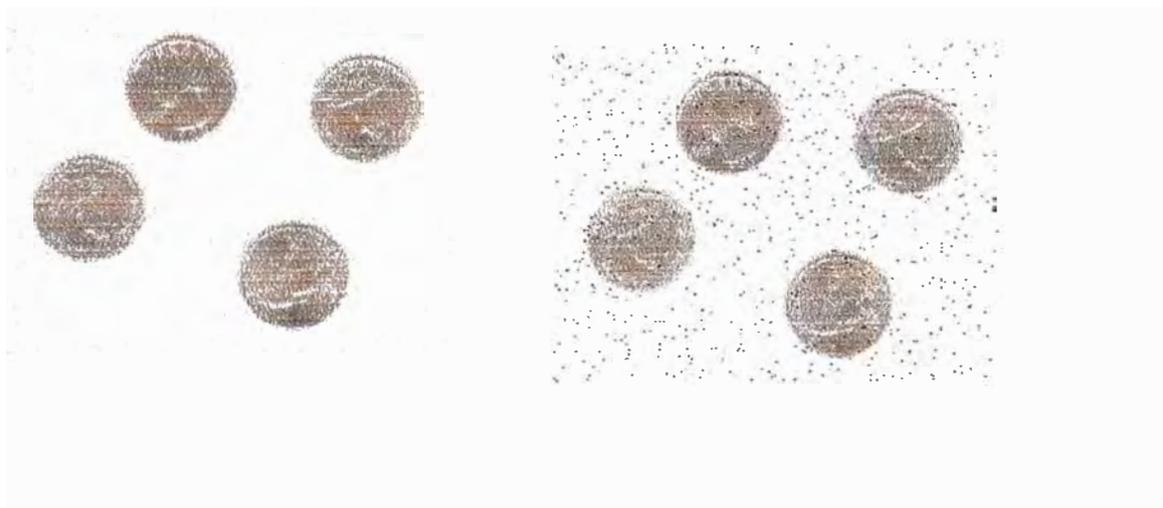
Permite remover ciertos tipos de ruido. Ciertos filtros, tales como filtros promedios o gaussianos son apropiados para este propósito. Por ejemplo, un filtro promedio es útil para quitar el ruido de grano de una fotografía. Porque cada pixel se pone al promedio de los pixeles en su vecindad, variaciones locales causadas por el grano son reducidas.

##### **Filtrado Mediano**

Es similar a usar un filtro promedio, en eso cada pixel de imagen de salida es "el promedio" de los valores de los pixeles de la vecindad de la imagen de entrada correspondiente. Sin embargo, con la filtrado mediano, el valor de un el pixel de salida es determinado por la mediana de los pixeles de la vecindad. La mediana es mucho menos sensible que la media para valores extremos. La filtración del medio es por consiguiente buena y capaz quitar estos valores extremos sin reducir la agudeza de la imagen.

La función **medfilt2** implementa el filtrado mediano. El ejemplo siguiente hace comparaciones que usan un filtro promedio y **medfilt2** para quitar el ruido de sal y pimienta. Este tipo de ruido consiste en píxeles aleatorios que son negros o blancos (los extremos de los rangos de datos). En ambos casos el tamaño de la vecindad usado para filtrar es 3 por 3. Primero, leemos la imagen y adicionamos ruido:

```
I = imread('eight.tif');
J = imnoise(I,'salt & pepper',0.02);
imshow(I)
figure, imshow(J)
```



Luego filtre el ruido de la imagen y displaye los resultados. Notar que **medfilt2** hace un mejor trabajo, con un manchado de bordes.

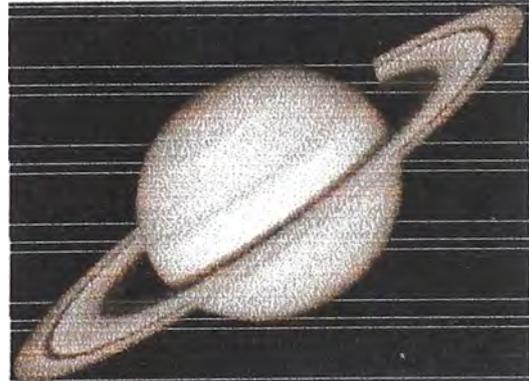
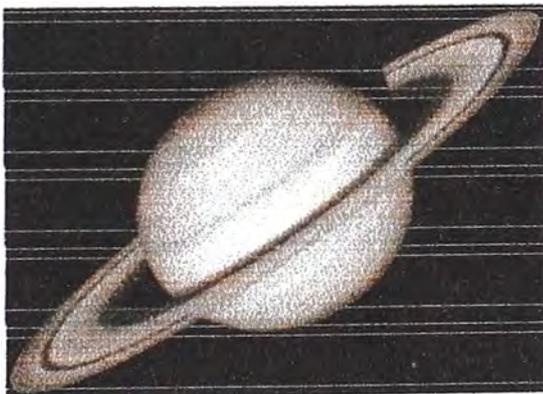
```
K = filter2(fspecial('average',3),J)/255;
L = medfilt2(J,[3 3]);
imshow(K)
figure, imshow(L)
```



#### 4.4.1. Filtrado Adaptivo

La función del `wiener2` aplica un filtro de Wiener (un tipo de filtro lineal) a una imagen, ajustándose a la medida de la variación de la imagen local. Donde la variación es grande, el `wiener2` realiza un suavizador pequeño. Donde la variación es pequeña, el `wiener2`, realiza un mayor suavizador. Esta aproximación produce a menudo mejores resultados que la filtración lineal. El filtro adaptativo es más selectivo que un filtro lineal comparable el `wiener2` trabaja mejor cuando el ruido es ruido aditivo de potencia constante ("blanco") como el ruido de Gaussiano. El ejemplo debajo de aplica el `wiener2` a una imagen de Saturno que ha tenido el ruido de Gaussian adicionado

```
l = imread('saturn.tif');
J = imnoise(l,'gaussian',0,0.005);
K = wiener2(J,[5 5]);
imshow(J)
figure, imshow(K)
```



## CAPITULO V APLICACIONES EN MATLAB PARA EL PROCESAMIENTO DE IMAGENES

### 5.1. Imágenes en Matlab

Una imagen en MATLAB es una matriz  $m$  por  $n$  de mosaicos de color, que se define mediante una matriz de imagen  $m$  por  $n$ . Las entradas de la matriz de imagen son números de color (índices de color) del mapa de color. Por ejemplo, consideremos una matriz 3 por 3:

$$w = [1 \ 2 \ 3; \ 2 \ 3 \ 2; \ 3 \ 2 \ 1]$$

y un mapa de color asociado de longitud 3:

$$\text{Mapa} = [1 \ 0 \ 0; \ 0 \ 1 \ 0; \ 0 \ 0 \ 1]$$

donde la primera fila (primera tripleta) es rojo, la segunda verde y la última azul. Entonces,

```
w = [1 2 3; 2 3 2; 3 2 1]
mapa = [1 0 0; 0 1 0; 0 0 1]
image (w)
Colormap (mapa)
```

producirá un patrón de mosaicos de color de 3 por 3 así:

```
Rojo Verde azul
Verde azul verde
Azul verde rojo
```

Aquí el orden de los mosaicos es el mismo que en la matriz (el primer índice de  $w$  se incrementa hacia abajo). Cabe señalar, empero, que si está vigente `hold on`, el orden vertical de la imagen se invierte (el primer índice se incrementa hacia arriba).

La imagen también puede exhibirse con `pcolor`. A continuación daremos más detalles sobre imagen y `pcolor`.

**Imagen:** `image` exhibe una imagen en la ventana de figura. Sus formatos son:

`Image(w)`

`Image(x, y, w)`

`h=image(w)`

El primer formato, `Image(w)`, exhibe la matriz  $w$  como una imagen. Cada elemento de  $w$  especifica el color de un mosaico rectangular de la imagen.

El segundo formato, `image(x, y, w)`, sólo resulta útil si es necesario imprimir marcas de escala con valores de coordenadas, donde  $x$  e  $y$  son vectores y especifican la rotulación de los ejes  $x$  y  $y$ . Sólo se utilizan el primero y el último elementos de  $x$  y  $y$  para los límites del eje. Los comando `image(w)` e `image(x, y, w)` producen exactamente la misma imagen excepto que las marcas de escala son alteradas por la segunda.

El tercer formato, `h = image(w)`, devuelve un mango de objetos asociados a `image`. Si desea ver una lista de las propiedades de los objetos de imagen y sus valores actuales, ejecute `set(h)`. Es posible añadir parámetros para especificar propiedades adicionales de la imagen colocándoles después de  $w$  en los argumentos del comando `image`. Este comando utiliza un ángulo de perspectiva especial de  $[0,90]$ ; es decir, el primer índice de  $w$  ( $i, j$ ) se incrementa verticalmente hacia abajo, y el segundo lo hace horizontalmente. Sin embargo, si está vigente `hold on`, el primer índice se incrementará hacia arriba. Los comandos `shading` y `caxis` no funcionan con `image`.

**Pseudocolor:** El comando `pcolor` gráfica con un patrón de tablero de ajedrez (pseudocolor). Los formatos son:

```
pcolor (w)
pcolor (x, y, w)
h= pcolor (w)
```

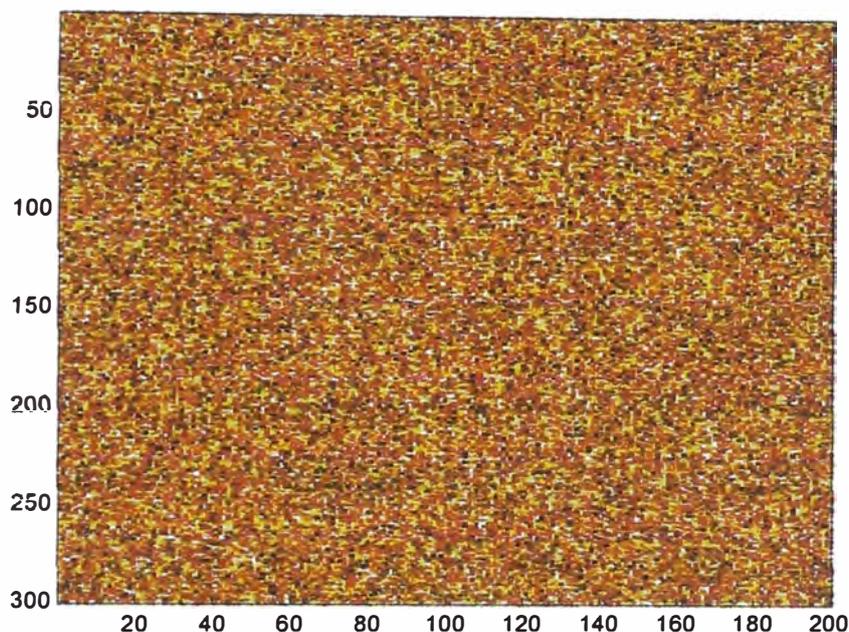
El comando `pcolor (w)` exhibe una gráfica en pseudocolor o “tablero de ajedrez” de la matriz `w`. La primera fila de `w` se convierte en la base de la figura (el inverso de `image (w)` a menos que se utilice `hold on`). Si se aplica `wiew (0,-90)`, el orden vertical será idéntico al de `image`. Los valores de los elementos de `w` especifican el color de cada celda de la gráfica. Los comandos `shading` y `caxis` son aplicables a `pcolor`. De hecho, `shading faceted` es el modo por omisión, en el que cada celda tiene un color constante y no se utilizan ni la última fila ni la última columna de `w`. Además, los mosaicos quedan separados por líneas negras. Si el tamaño de los mosaicos es pequeño, las líneas negras podrían dominar la imagen y oscurecerlas. Las líneas negras pueden eliminarse con `shading flat`. Con otra opción, `shading interp`, los colores de un mosaico cambian continuamente por interpolación bilineal, de modo que la imagen se suaviza. Con el modo por omisión de `caxis`, a los elementos más pequeños y más grande de `w` se les asigna el primero y el último de los colores de la tabla de colores; los colores para el resto de los elementos de `w` se determinan mediante una búsqueda de tabla en el resto del mapa de colores. Sin embargo, el comando `caxis` permite cambiar el intervalo de colores.

He aquí una forma fácil de experimentar con `image` y `pcolor`: trazar una imagen de una matriz aleatoria ejecutando el siguiente guión:

Listado D.1

```
% rand_im plots an image of random matrix.
disp('Input matrix size:')
m=input('m= ');
n=input('n= ');
W = ceil(64*rand(m,n));
% Generates a m-by-n random matrix.close
```

```
colormap(hot)
set(gcf, 'NumberTitle','off','Name',...
    'rand_im: image of a random matrix')
image(W);
```



La multiplicación de la matriz aleatoria por 64 es necesaria porque los números aleatorios producidos por rand están entre 0 y 1, mientras que los índices de color están entre 1 y 64. La operación  $\text{ceil}(64 * \text{rand}(m, n))$  sirve para producir una matriz de números aleatorios entre 1 y 64. Pruebe con un número pequeño de mosaicos, digamos 5 por 5 ( $m = n = 5$ ) o 20 por 30 ( $m = 20, n = 30$ ) primero, y luego con una mayor, digamos 50 por 50. Si tiene una versión profesional de MATLAB, intente 200 por 300. Una vez que haya terminado de experimentar con el mapa de color hot, puede cambiarlo por cualquier otro mapa de color. Pruebe colormap jet, colormap cool, colormap flag desde la ventana de comandos. A diferencia de pcolor, los comandos caxis, shading flat y shading interp no pueden aplicarse a image.

Una vez que entienda lo que funciona y lo que no con `image`, cambien este comando a `pcolor` en el listado D.1 y ejecútelo. A diferencia de `image`, los comandos `caxis`, `shading` y `view` sí funcionan. Por ejemplo, `view(0, -90)` invierte horizontalmente la imagen de `pcolor`. Por otro lado, `caxis([-100, 100])` reduce la amplitud de la gama de color de la imagen. Aplique `shading interp` y también `shading flat`. Con este último comando observará que las líneas negras que separan los mosaicos desaparecen y la imagen queda mucho más brillante que con la opción por omisión `shading faceted`. Con `shading interp`, la imagen se suaviza.

## 5.2. Almacenamiento y Carga de Imágenes en Formato Estándar

Los datos estándar de imágenes en MATLAB consisten en una matriz de imagen `w` y un mapa de color asociado `map`. Para guardar una imagen en el formato estándar, escriba

Save nombre de archivo `w` mapa

**Si desea cargar el archivo, escriba**

`[w, mapa] = load nombre de archivo`

## 5.3. Exportación e Importación de Archivos de Imágenes

Podemos tener muchas razones para transportar archivos de imágenes entre MATLAB y otro software. He aquí algunos ejemplos comunes de ellas:

- (a) Necesitamos reprocesar con MATLAB imágenes de pintura, dibujo o fotográficas creadas con otro software.
- (b) Las figuras o imágenes creadas con MATLAB pueden exhibirse o reprocesarse con otro software. Podemos desarrollar la geometría de los objetos con MATLAB, aprovechando las múltiples opciones de funciones matemáticas y herramientas de interpolación. Por otro lado, editar una imagen existente o retocarla con colores puede ser más fácil en alguna otra aplicación de software.
- (c) En muchos programas no es posible capturar gráficos en formato `postscript`. Podríamos importar imágenes en MATLAB con objeto de crear un archivo `postscript`.

(d) En algunas situaciones la impresión de imágenes puede ser más fácil con otro software de procesamiento de imágenes o de pintura que directamente desde MATLAB.

A menos que se utilice la Image Processing Toolbox, el único formato en que podemos importar y exportar archivos de imagen de y a otro software es el formato de imagen en bruto.

La imagen en bruto puede introducirse a MATLAB con `imread`. El programa Alchemy puede realizar conversiones entre casi cualquier formato de imagen y el formato en bruto. Un archivo gif, digamos `imagen04.gif`, se puede convertir al formato en bruto con

```
Alchemy -r -c 128 -z 4 imagen04.gif
```

que crea el archivo `image04.raw`, donde

- r: produce un archivo "en bruto"
- c 128: limita la longitud del mapa de color a 128 colores;
- z 4: ordena el mapa de color por luminancia.

La imagen de este archivo puede exhibirse en MATLAB con

```
[w, map] = imread ('imagen04.raw');
colormap (map)
image (w)
```

Si puede obtener acceso a Alchemy directamente desde MATLAB (mediante una trayectoria apropiada), los procesos anteriores se pueden combinar en un solo guión:

```
Alchemy -r -c 128 -z 4 image04.gif
[w, map] = imread ('image04.raw');
colormap (map)
image (w)
```

El inverso de `imread` es `imwrite`. Podemos convertir datos de imagen de MATLAB en un archivo gif con

```
[w, mapa] = load datos_imagen
imwrite (w, mapa, 'nomarchv.raw')
! alchemy -g nomarchv.raw nomarchv.gif
% convierte a GIF
```

## 5.4. Manejo de Elementos del Procesamiento de Imágenes

Como una imagen consiste en una matriz de índices de color y un mapa de color asociados, la imagen puede alterarse de muchas formas modificando la matriz y el mapa de color. El cambio de color y el cambio de contraste son las operaciones de proceso de imágenes más elementales. El solo cambio del esquema de color puede proporcionar información distinta del mismo archivo de imagen. La Image Processing Toolbox cuenta con muchas herramientas para procesar imágenes. En el resto de esta sección describiremos aspectos fundamentales de unas cuantas técnicas elementales de procesamientos de imágenes.

### Ejemplo D.2 Extracción de bordes y difusión

La figura D.3 muestra una imagen de círculos de color traslapados trazada con el listado D.2.

**Figura D.2** Tres discos.

#### Listado D.2

**disk.ptn. m**

```
% disk_ptn.m Plots three disks by pcolor.
close
set(gcf, 'NumberTitle','off','Name', 'disk_ptn; L_d2')
clear;clf
map=jet;
colormap(map)
disp 'For Student Edition users, ni < 25'
%ni=input('ni = ')
ni=100;
nj= ni*1.20;
ic1=ni/2; ic2=ni/2; ic3=ni/2;
jc1=nj/4; jc2=nj/2; jc3=nj/4*3;
W=ones(ni,nj);
ni20=ni/4;
for i=1:ni
    for j=1:nj
        r1 = sqrt((i-ic1)^2 + (j-jc1)^2);
```

```

r2 = sqrt((i-ic2)^2 + (j-jc2)^2) ;
r3 = sqrt((i-ic3)^2 + (j-jc3)^2);
if r1 < ni20 W(i,j)=20; end
if r2 < ni20 , W(i,j)=40; end
if r3 < ni20 , W(i,j)=60; end
if r1< ni20 & r2 < ni20, W(i,j)=30; end
if r2< ni20 & r3 < ni20, W(i,j)=50; end
end
end
pcolor(W);
shading flat;
save disk_d W map ; axis('off')
text(ni/10,ni/10,'Disk pattern','FontSize',[18])

```

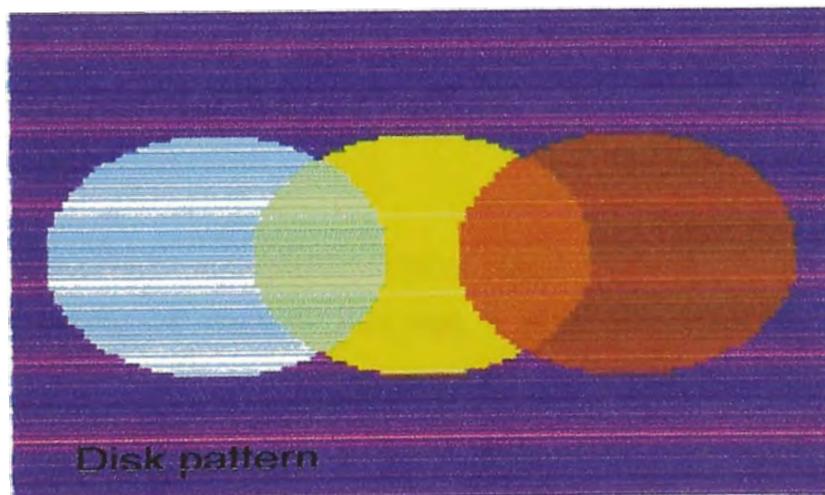


Figura D 2 : Tres discos

Los bordes de los discos pueden identificarse por cambios repentinos del índice de color en la matriz de la imagen  $w$ . Por ejemplo,

$$C(i, j) = [w(j, i) - w(j - 1, i)]^2 + [w(j, i) - w(j + 1, i)]^2 + [w(j, i) - w(j, i - 1)]^2 + [w(j, i) - w(j, i + 1)]^2$$

Será cero si el mosaico ( i , j ) no está adyacente aun borde, y distinto de cero si no lo está. La imagen ilustrada en la figura D.3 se dibujó con el listado D.3.



**Figura D.3** Bordes de tres discos

### Listado D.3

#### Disk\_edg.m

```
% disk_edg.m plots edges extracted from three disks.
% set(gcf, 'NumberTitle','off','Name', 'disk_edg; L_d3')
clear,clf
load disk_d %reads W and map
[ni,nj]=size(W);
colormap(map)
c = zeros(size(W));
for i=2:ni-1
for j=2:nj-1
c(i,j)= (W(i-1,j)-W(i,j))^2 + (W(i+1,j)-W(i,j))^2 ...
+ (W(i,j-1)-W(i,j))^2 + (W(i,j+1)-W(i,j))^2;
```

```

    if c(i,j) > 0, c(i,j) = 55; end
end
end
image(c);axis('off')
text(ni/10,ni/10,'Extracted edge pattern','FontSize',[18])
save edge_d c map

```

El patrón de los bordes puede hacerse difuso aplicando un operador de difusión, o bien resolviendo la siguiente ecuación:

$$(4 + \alpha) d_{j,i} = d_{j-1,i} + d_{j+1,i} + d_{j,i-1} + d_{j,i+1} + c(j,i) \quad (D.1)$$

donde  $\alpha$  es un parámetro para controlar el alcance del efecto de difusión. Cuanto más alto sea  $\alpha$ , más corta será la distancia de difusión. El siguiente es un esquema de resolución interactivo basado en el método de Gauss – Seidel:

$$(4 + \alpha) d_{j,i}^{(t)} = d_{j-1,i}^{(t-1)} + d_{j+1,i}^{(t-1)} + c(j,i)$$

donde  $t$  es el número de iteración. En el caso del procesamiento de imágenes no es necesaria una convergencia estricta- De hecho, un número reducido de pasos de iteración logra el efecto de difusión.

Una variante del procesamiento anterior es el efecto de *difusión convectiva* producido por

$$(2 + \alpha) d_{j,i}^{(t)} = d_{j+1,i}^{(t-1)} + c(j,i) \quad (D.3)$$

El esquema anterior difunde la imagen sólo en una dirección a partir de los bordes. El listado D.4 exhibe el patrón difuso de la figura D.5 La versión coloreada se encuentra en las líneas de color.

#### Listado D.4

##### Edge\_dif.m

```

% edge_dif.m Bordes difusos de tres discos
clear, clf
load edgeda

```

```
[ni, nj] = size ?
```

```
d = c;
```

**Figura D.5** Tres discos después de difusión

```
Colormap (map)
```

```
for iter = 1:7
```

```
for l= 2: ni - 1
```

```
for j= 2: nj -1
```

```
    d(l,j) = (d(l-1, j) + d(l, j) + d(l, j - 1) + d (l,j) + c (l, j) / (4+0.1));
```

```
end
```

```
end
```

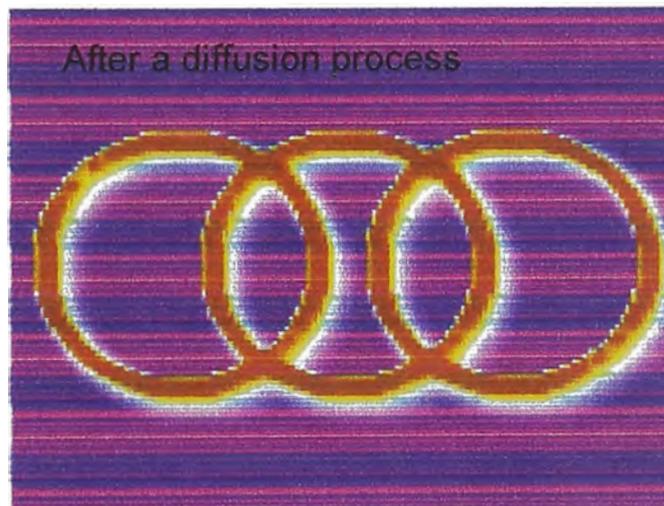
```
end
```

```
image (d)
```

```
Text (ni/10, ni/10, ' Después de un proceso de difusión ', 'FontSize'[18])
```

```
Axis ( 'off')
```

**Figura D.5:** Tres discos después de la difusión



## CONCLUSIONES

1. El proceso de imágenes incluye un conjunto diverso de métodos para la manipulación de las imágenes digitales. Incluyen la modificación y el realce (aumentando el contraste o cambiando la dimensión de una variable de la imagen) y el análisis (considerando objetos deseados dentro de la imagen).
2. Las imágenes digitales se representan como matrices de dos dimensiones o "píxeles" que representan los grados de brillo (rojos, verdes, y azules para una imagen de color) en localizaciones uniformemente espaciadas.
3. Muchos métodos de procesamiento de imágenes pretenden ser análogos al proceso visual humano. La facilidad evidente con la cual vemos es engañosa puesto que de mucho de ella no somos conscientes.
4. Los algoritmos en computadora para el procesamiento de imágenes son complejos y funcionan mejor cuando las señales de imágenes están libres de ruido. El ruido generado en la adquisición de imágenes debe ser eliminado mediante un filtrado.
5. Las técnicas que involucran la mejora de la imagen se definen a veces objetivamente (aumentando la relación señal a ruido) y a veces subjetivamente (permitiendo ver ciertos rasgos de la imagen, modificando los colores o intensidades).
6. El procesamiento de imágenes, va mejorando debido al desarrollo de computadoras cada vez más veloces, con alta capacidad de procesamiento y precios cada vez menores, con mayores capacidades de almacenamiento de información y la existencia de software amigable para el procesamiento como el MATLAB.

## BIBLIOGRAFIA

1. B. Saleh, M. C. Teich, Fundamentals of photonics, John Wiley & Sons.
2. K. Castleman, Digital imaging processing, Prentice Hall.
3. Rafael C. González, Richard E. Woods, Digital Image Processing, Addison Wesley, 1993.
4. Anil K. Jain., Fundamentals of Digital Image Processing, Prentice Hall 1998.
5. Robert J. Schalkoff., Digital Image Processing and Computer Vision, John Wiley and Sons, 1989.
6. Friederich M. Wahl, Digital Image Signal Processing, Artech House.
7. Pratt, Digital Image Processing, John Wiley & Sons.
8. Ferdinand van der Heijden, Image Based Measurement Systems, John Wiley & Sons. (1994).
9. Rafael C. González & Richard E. Woods, Procesamiento Digital de Imágenes., Editorial Addison – Wesley, 1996
10. Scott E. Umbaugh, Computer Vision and Image Processing, Editorial Prentice Hall
11. Robert M. Haralick & Linda G. Shapiro, Computer and Robot Vision. Vol I y II, Editorial Addison – Wesley
12. Jain. A.K., Fundamentals of Digital Image Processing, Editorial Prentice Hall
13. W. K. Pratt, Digital Image Processing, Editorial Wiley