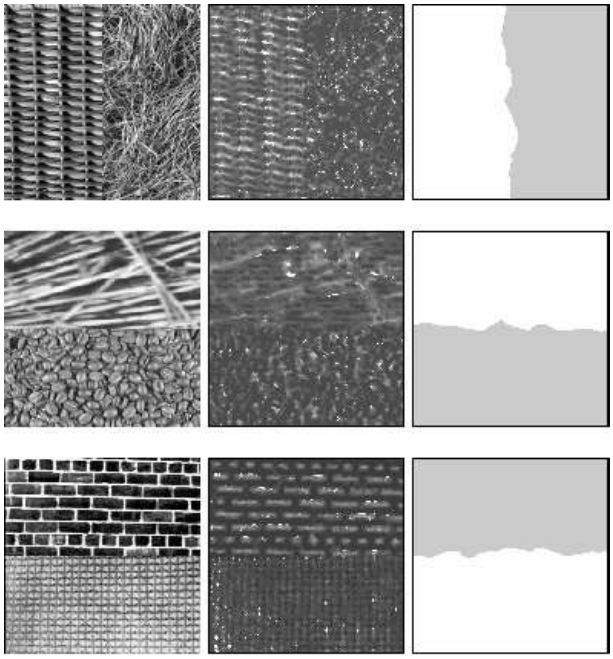
Visión Computacional



L. Enrique Sucar

Instituto Nacional de Astrofísica, Óptica y Electrónica

Puebla, México

Giovani Gómez

Helmholtz Zentrum Munchen

Neuherberg, Alemania

Prólogo

Según Aristóteles, Visión es *saber que hay y dónde mediante la vista*, lo cual es escencialmente válido. Nuestra vista y cerebro identifican, a partir de la información que llega a nuestros ojos, los objetos que nos interesan y su posición en el ambiente, lo cual es muy importante para muchas de nuestras actividades. La *Visión Computacional* trata de alguna forma de emular esta capacidad en las computadoras, de forma que mediante la interpretación de las imágenes adquiridas, por ejemplo con una cámara, se puedan reconocer los diversos objetos en el ambiente y su posición en el espacio.

La facilidad con la que "vemos", llevó a pensar a los primeros investigadores en inteligencia artificial, por 1960, que hacer que una computadora interpretara imágenes era relativamente fácil, Pero no resultó así, y muchos años de investigación han demostrado que es un problema muy complejo. Sin embargo, en los últimos años hay avances considerables básicamente por 3 factores:

- El desarrollo tecnológico en las capacidades de procesamiento y de memoria en las computadoras, que facilita el almacenamiento y procesamiento de las imágenes.
- Los avances teóricos en los principios y algoritmos para el procesamiento y análisis de imágenes.
- La creciente necesidad del procesamiento automático de imágenes, que se capturan y almacenan en grandes cantidades en diversos dominios, como en medicina, seguridad, tránsito de vehículos, etc.

Este creciente interés en el desarrollo de sistema de visión automáticos ha creado una necesidad de la formación de especialistas en este campo, y por consiguiente a su incorporación como un curso común en los posgrados e incluso licenciaturas en computación, informática y electrónica. Sin embargo, existen pocos textos, en particular en castellano, que presenten una introducción general a visión computacional. La mayor parte de los libros tienen un enfoque más hacia procesamiento de imágenes que hacia visión. Aunque están relacionados, hay una diferencia fundamental entre ambos enfoques: procesamiento de imágenes trata sobre como *mejorar* una imagen para su interpretación por una persona; mientras que visión computacional busca la interpretación de las imágenes por la computadora. Otros libros se centran en aspectos particulares de visión, como la visión tridimensional, por lo que no presentan un panorama general del área como se requiere en un curso introductorio.

Este libro presenta una introducción general a visión por computadora, basado en un esquema sistemático en el cual el proceso de visión computacional se puede dividir en 3 grandes etapas:

- Procesamiento de nivel bajo - se trabaja directamente con las imágenes para extraer propiedades como orillas, gradiente, profundidad, textura, color, etc.
- Procesamiento de nivel intermedio - consiste generalmente en agrupar los elemento obtenidos en el nivel bajo, para obtener, por ejemplo,contornos y regiones, generalmente con el propósito de *segmentación*.
- Procesamiento de alto nivel - consiste en la interpretación de los entes obtenidos en los niveles inferiores y se utilizan modelos y/o conocimiento *a priori* del dominio.

Aunque estas etapas no son indispensables en todo sistema de visión y tampoco necesariamente secuenciales, permiten dividir los principales temas en una forma congruente y natural para su enseñanza.

De acuerdo a lo anterior, los capítulos están organizados de la la siguiente manera:

Cada capítulo presenta una intruducción general al tema y las principales técnicas básicas que se han desarrollado, incluyendo ejemplos con imágenes. Al final del capítulo se hace una breve reseña histórica de la investigación en dicho aspecto de visión, incluyendo referencia adicionales para quien desee ir más alla. Se incluye una lista de problemas y de proyectos prácticos por capítulo. Al final se encuentra listada toda la bibliografía que se menciona en el texto.

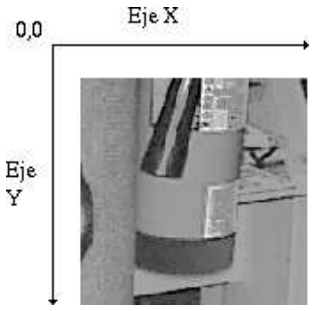
El libro esta orientado a un curso semestral introductorio de visión computacional, ya sea de posgrado o de últimos semestres de licienciatura. No existe un prerrequisito particular, salvo las bases generales de matemáticas (algebra, cálculo, probabilidad) y de computación (programación en algun lenguaje, organización de computadoras).

L. Enrique Sucar
Giovani Gómez

Contenido

Índice de Figuras

Índice de Tablas



Capítulo 1

Introducción

1.1 ¿Qué es visión?

Visión es la ventana al mundo de muchos organismos. Su función principal es reconocer y localizar objetos en el ambiente mediante el procesamiento de las imágenes. La *visión computacional* es el estudio de estos procesos, para entenderlos y construir máquinas con capacidades similares. Existen varias definiciones de visión, entre éstas podemos mencionar las siguientes.

- "Visión es saber que hay y dónde mediante la vista" (Aristóteles).
- "Visión es recuperar de la información de los sentidos (vista) propiedades válidas del mundo exterior", Gibson [25].
- "Visión es un *proceso* que produce a partir de las imágenes del mundo exterior una *descripción* que es útil para el observador y que no tiene información irrelevante", Marr [77].

Las tres son escencialmente válidas, pero la que tal vez se acerca más a la idea actual sobre visión computacional es la definición de Marr. En esta definición hay tres aspectos importantes que hay que tener presentes: (i) visión es un proceso computacional, (ii) la descripción a obtener depende del observador y (iii) es necesario eliminar la información que no sea útil (reducción de información).

Un área muy ligada a la de visión computacional es la de *procesamiento de imágenes*. Aunque ambos campos tienen mucho en común, el objetivo final es diferentes. El objetivo de procesamiento de imágenes es mejorar la calidad de las imágenes para su posterior utilización o interpretación, por ejemplo:

- remover defectos,
- remover problemas por movimiento o desenfoque,
- mejorar ciertas propiedades como color, contraste, estructura, etc.
- agregar "colores falsos" a imágenes monocromáticas.

En la figura 1.1 se ilustra el enfoque de procesamiento de imágenes, en el cual se obtiene una imagen "mejor" para su posterior interpretación por una persona.

El objetivo de la visión computacional es extraer características de una imagen para su descripción e interpretación por la computadora. Por ejemplo:

- determinar la localización y tipo de objetos en la imágen,

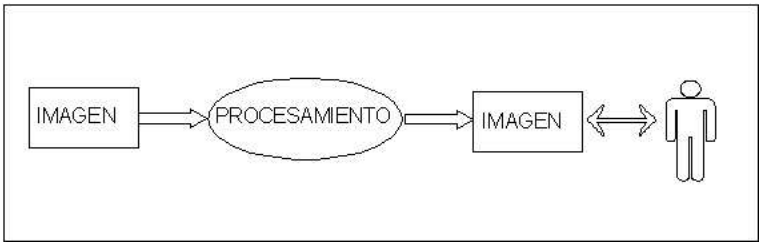


Figura 1.1: Esquema general del procesamiento de imágenes. Su función principal es presentar la *misma* imágen resaltando e ignorando ciertas características. Observese que la entrada y salida son imágenes.

- contruir una representación tridimensional de un objeto,
- analizar un objeto para determinar su calidad,
- descomponer una imágen u objeto en diferentes partes.

En visión se busca obtener descripciones útiles para cada tarea a realizar. La tarea demandará modificar ciertos atributos, ver figura 1.2.

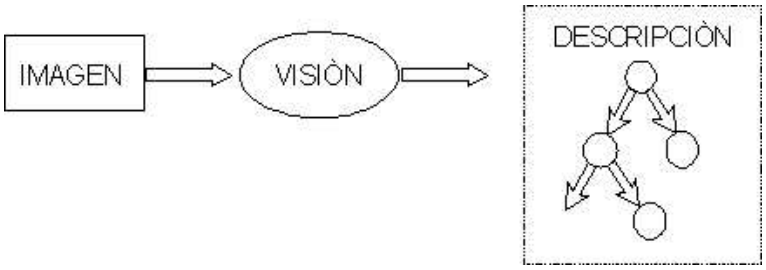


Figura 1.2: Esquema general de visión por computadora. La imagen de entrada es procesada para extraer los atributos, obteniendo como salida una descripción de la imagen analizada.

En la figura 1.3 se muestra un ejemplo de procesamiento de imágenes. La tarea a realizar es *mejorar* la imagen de entrada, la cual es obscura. La imagen de salida es escencialmente la *misma* pero de mejor calida o "más útil". La figura 1.4 ilustra la diferencia entre procesamiento de imágenes y visión; notese que la imagen muestra ciertas descripciones importantes, como los números, que previamente fueron detectados. La salida de este sistema de visión se complementa con un módulo de reconocimiento de patrones, es decir, "saber" que letras y números contiene la placa.

Figura 1.3: Aumento de contraste: (a) imagen obscura debido a que su rango de grises es reducido, (b) ecualización del rango de grises.

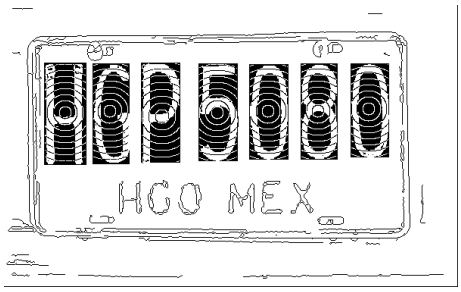


Figura 1.4: Reconocimiento de caracteres en base a su codificación radial.

Actualmente existen múltiples aplicaciones prácticas de la visión computacional, entre éstas podemos mencionar las siguientes:

- Robótica móvil y vehículos autónomos. Se utilizan cámaras y otros tipos de sensores para localizar obstáculos, identificar objetos y personas, encontrar el camino, etc.
- Manufactura. Se aplica visión para la localización e identificación de piezas, para control de calidad, entre otras tareas.
- Interpretación de imágenes aéreas y de satélite. Se usa procesamiento de imágenes y visión para mejorar las imágenes obtenidas, para identificar diferentes tipos de cultivos, para ayudar en la predicción del clima, etc.
- Análisis e interpretación de imágenes médicas. La visión se aplica para ayudar en la interpretación de diferentes clases de imágenes médicas como rayos-X, tomografía, ultrasonido, resonancia magnética y endoscopía.
- Interpretación de escritura, dibujos, planos. Se utilizan técnicas de visión para el reconocimiento de textos, lo que se conoce como *reconocimiento de caracteres*. También se aplica a la interpretación automática de dibujos y mapas.
- Análisis de imágenes microscópicas. El procesamiento de imágenes y visión se utilizan para ayudar a interpretar imágenes microscópicas en química, física y biología.
- Análisis de imágenes para astronomía. Se usa la visión para procesar imágenes obtenidas por telescopios, ayudando a la localización e identificación de objetos en el espacio.
- Análisis de imágenes para compresión. Aunque la compresión de imágenes ha sido tradicionalmente una subárea del procesamiento de imágnes, recientemente se están desarrollado técnicas más sofisticadas de compresión que se basan en la interpretación de las imágenes.

1.2 Formación y representación de la imágen

La formación de la imágen ocurre cuando un sensor (ojo, cámara) registra la radiación (luz) que ha interactuado con ciertos objetos físicos, como se muestra en la figura 1.5. La imagen obtenida por el sensor se puede ver como una función bidimensional, donde el valor de la función corresponde a la intensidad o brillantez en cada punto de la imagen (imágenes monocromáticas, conocidas como imágenes en "blanco y negro"). Generalmente, se asocia un sistema coordenado (x, y) a la imagen, con el origen en el extremo superior izquierdo, ver figura 1.6.

Una función de la imagen es una representación matemática de la imágen. Esta es generalmente una función de dos variables espaciales (x, y):

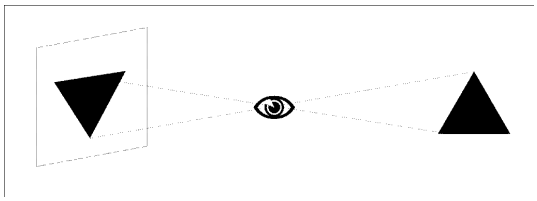


Figura 1.5: Formación de la imagen.

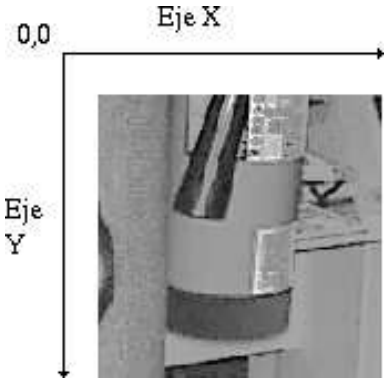


Figura 1.6: Ejemplo los ejes (x, y) en una imagen.

$$I = f(x, y) \tag{1.1}$$

Donde f representa el nivel de brillantez o intensidad de la imagen en las coordenadas (x, y). Si representamos estas funciones gráficamente, se tienen 3 dimensiones: dos que corresponden a las coordenadas de la imagen y la tercera a la función de intensidad, ver figura 1.7.

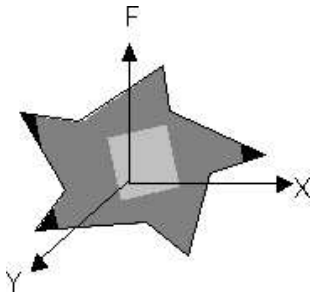


Figura 1.7: Representación matemática de una imagen: *f(x,y)*.

Una imagen multiespectral f es una función vectorial con componentes $(f_1, f_2, ..., f_n)$, donde cada una representa la intensidad de la imagen a diferentes longitudes de onda. Por ejemplo, una imágen a color generalmente se representa por la brillantez en tres diferentes longitudes de onda:

$$f(x, y) = [f_{rojo}(x, y), f_{azul}(x, y), f_{verde}(x, y)] \tag{1.2}$$

Una *imagen digital* es una imagen que ha sido discretizada tanto en valor de intensidad (f) como espacialmente, es decir que se ha realizado un muestreo de la función continua. Este muestreo se representa matemáticamente mediante la multiplicación de la función con un arreglo bidimensional de funciones delta:

$$f_s(x, y) = \int \int_{-\infty}^{\infty} f(x, y), \delta(x - x_0, y - y_0), dx, dy \tag{1.3}$$

Donde cada valor de intensidad, $f_s(x, y)$, es mapeado o discretizado a un número, por ejemplo un número entre 0 y 255. Entonces una imagen digital monocromática puede ser representada por una matriz de NxM, donde cada valor es un número que representa el nivel de intensidad del punto correspondiente de la imagen. Cada punto se conoce como *pixel* (del inglés, *picture element*).

1.2.1 Proyección de la Imagen

La proyección puntual es la transformación de la imagen que se presenta al pasar a muchos de los dispositivos visuales, incluyendo nuestros ojos y una cámara. La aproximación más simple a este fenómeno es el modelo de la "cámara de agujero de alfiler" (*pinhole camara*) que consiste en proyectar todos los puntos de la imágen a través del un punto al plano de la imagen. De esta forma, un punto (X, Y, Z) en el espacio, se proyecta a un punto (x, y) en el plano de la imagen. El plano de la imagen se encuentra a una distancia f del "agujero" o lente de la cámara, la cual se conoce como distancia focal. En la figura 1.8 se ilustra en forma simplificada el modelo geométrico de una cámara.

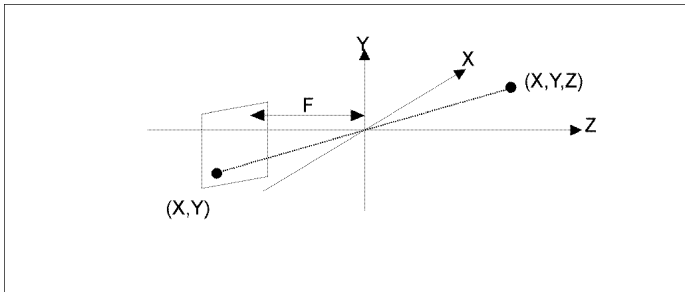


Figura 1.8: Modelo geométrico de la cámara. El plano de la imagen esta dado por los ejes x, y. z es la perpendicular del plano $x - y$ a la cámara, y F es la distancia del punto de proyección al plano de la imágen (distancia focal).

Para evitar la inversión de la imagen y simplificar las matemáticas se considera el plano de la imagen del mismo lado que la imagen y $z = 0$ sobre dicho plano, como se puede ver en la figura 1.9.

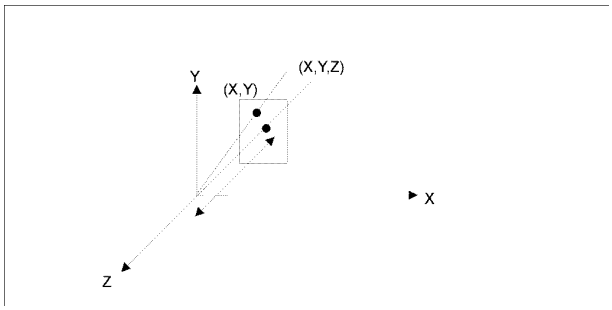


Figura 1.9: Modelo geométrico equivalente.

Consideremos, inicialmente, sólo la proyección respecto a la coordenada Y del punto, como se ilustra en la figura 1.10. De acuerdo a este modelo, el tamaño relativo de un objeto en la imagen depende de la distancia al plano de la imagen (z) y la distancia focal (f). Por triángulos semejantes obtenemos:

$$\frac{y}{f} = \frac{Y}{(F - Z)} \tag{1.4}$$

De donde $y = \frac{fY}{(F-Z)}$.

En forma similar obtenemos la ecuación para x. Entonces la transformación para la llamada *proyección prespectiva* es:

$$(x, y) = [\frac{fX}{(F - Z)}, \frac{fY}{(F - Z)}] \tag{1.5}$$

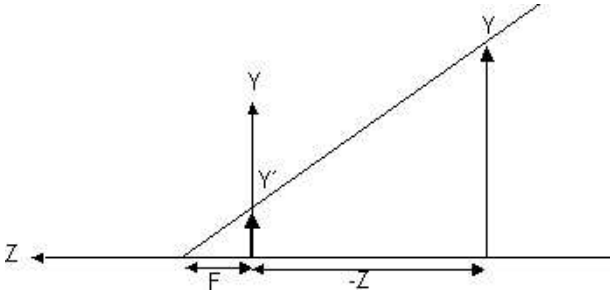


Figura 1.10: Proyección en Y

Si consideramos el punto de vista en el infinito (Z), obtenemos un caso especial denominado *proyección ortográfica*, que para el caso de la coordenada Y se muestra en la figura 1.11. En este caso la proyección de los puntos es paralela al eje de observación, Z, por lo que las coordenadas (x, y) de la imagen son iguales a las coordenadas (X, Y) en el espacio. Este tipo de proyección se puede utilizar como una aproximación práctica si la distancia entre la cámara y los objetos es muy grande en relación con el tamaño de los objetos.

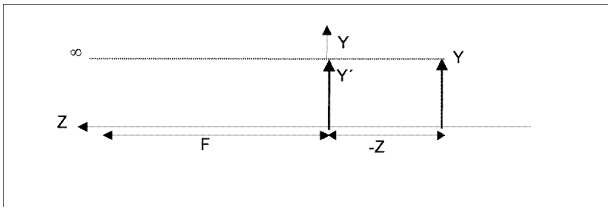


Figura 1.11: Proyección ortográfica.

1.2.2 Imágenes binoculares

Al proyectarse los objetos, de un espacio tridimensional a una imagen bidimensional se pierde la información de la distancia a la cámara o profundidad (eje Z) de cada punto. Una forma de tratar de recuperar esta información es mediante el uso de dos cámaras, en lo que se conoce como visión estéreo.

Si consideramos que tenemos dos cámaras separadas a una distancia conocida $2d$, tendremos dos imágenes de cada punto (X, Y). Utilizando sólo la coordenada Y, el modelo geométrico se puede ver en la figura 1.12.

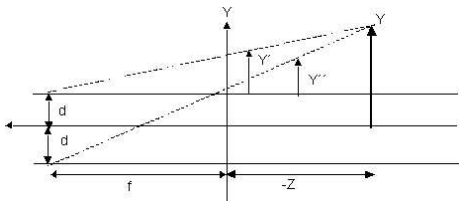


Figura 1.12: Imágenes binoculares.

Las ecuaciones para la proyección perspectiva de; modelo geométrico para dos cámaras son las siguientes:

$$y' = \frac{(Y-d)f}{(f-Z)} \tag{1.6}$$

$$y'' = \frac{(Y+d)f}{(f-Z)} \tag{1.7}$$

De donde podemos obtener el valor de Z:

$$Z = \frac{f-2df}{(y'-y'')} \tag{1.8}$$

De aquí podríamos pensar que el extraer información de profundidad es aparentemente simple teniendo un sistema con dos cámaras (estereo). Pero el problema, como veremos más adelante, es encontrar la correspondencia (*matching*) entre los puntos de las dos imágenes.

1.2.3 Reflectancia

La brillantez de cada punto en la imagen depende de las propiedades físicas del objeto a observar, así como también de las condiciones de iluminación presentes. La reflectancia depende del tipo de superficie, geometría, ángulo de incidencia de la fuente lumínica, color y demás propiedades intrínsicas del mismo objeto.

La intensidad que radía la fuente lumínica (I), en $watts/steradian$, se define como el flujo por ángulo sólido:

$$I = d\phi/d\omega \tag{1.9}$$

Y el flujo incidente (E) sobre un elemento dA del objeto es:

$$E = d\phi/dA \tag{1.10}$$

Donde:

$$d\omega = dA/r^2 \tag{1.11}$$

El flujo emitido por la superficie (L) depende de el flujo incidente y el ángulo respecto a la superficie del objeto:

$$L = d^2\phi/dAcos\theta d\omega \tag{1.12}$$

La brillantez (f) en la imagen va a ser proporcional a dicho flujo emitido por la superficie del objeto. La figura 1.13 ilustra en forma simplificada el fenómeno.

En general, la brillantez o intensidad de la imágen va a depender de 3 factores:

- La fuente lumínica.
- La geometría (posición de la fuente, objeto, cámara).

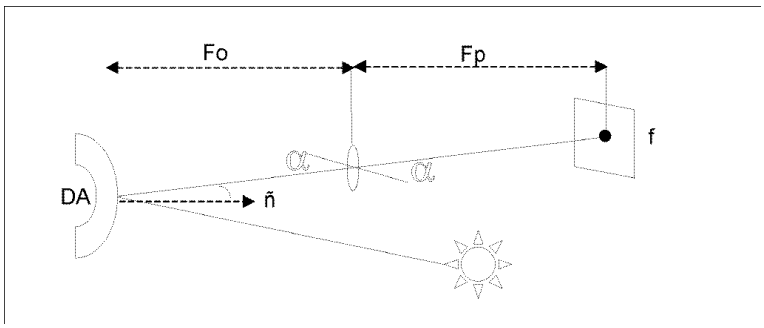


Figura 1.13: Reflectancia. La luz emitida se refleja en el objeto (DA), y es recibida por el sensor, generando cierta brillantez en la imagen (f). La brillantez depende de la intensidad de la fuente (I), el ángulo (α) del rayo con la normal (ñ) de la superficie, las propiedades de reflectancia del objeto y la distancia del objeto a la imagen ($Fo + Fp$).

- Las propiedades intrínsicas del objeto (reflectancia).

Existen dos tipos básicos de superficie:

- Mate (*lambertian*). Refleja la luz recibida en todas direcciones.
- Especular. Refleja la luz recibida en una sola dirección, la cual está en función del ángulo entre el rayo incidente y la normal a la superficie.

Las superficies del mundo real muestran una combinación de ambas.

1.2.4 Color

El color es un fenómeno perceptual relacionado con la respuesta humana a diferentes longitudes de onda del espectro visible (400 - 700 nm). Esto se debe a que existen tres tipos de sensores en el ojo que tienen una respuesta relativa diferente de acuerdo a la longitud de onda. Esta combinación de tres señales da la sensación de toda la gama de colores que percibimos. La figura 1.14 muestra en forma gráfica las diferentes respuestas relativas de los tres tipos de sensores (α, β, γ) respecto a la longitud de onda.

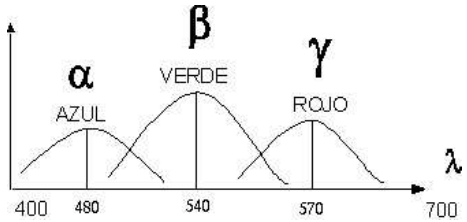


Figura 1.14: Respuesta en *nm* de los diferentes tipos de sensores al color.

Existen diferente formas de organizar o codificar los diferentes colores a partir de componentes básicas, lo que se conoce como *espacios de color*. Los modelos RGB y HSI son un ejemplo de tales espacios o modelos de color.

Modelo RGB

El modelo RGB se basa en los tres sensores humanos, considerando que todos los colores son una combinación de tres colores básicos o primarios: R (rojo), G (verde), B (azul). Generalmente los

componentes se normalizan, obteniendo:

- r = R / (R + G + B)
- g = G / (R + G + B)
- b = B / (R + G + B)

Se pueden visualizar a todos los colores dentro de un triángulo, ver figura 1.15, en cuyos vértices se encuentran los componentes primarios, R, G, B. Todos los demás colores, dentro del triángulo, se pueden obtener como una combinación lineal de los primarios. El color blanco se encuentra en el centro del triángulo, con igaul proporción de cada color primario. La televisión y las pantallas de computadora se basan en este modelo para generar toda la gama de colores.

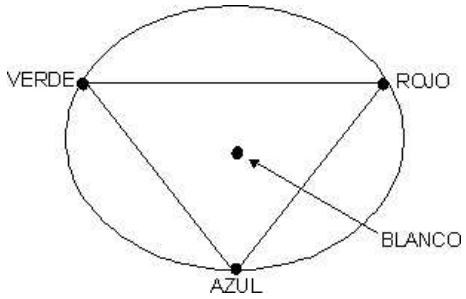


Figura 1.15: Diagrama cromático.

Modelo HSI

Se considera que el modelo HSI es el que mejor aproxima a la percepción humana. El modelo HSI codifica el color en tres componentes:

- I - intensidad (brillantez).
- H - croma (*Hue*).
- S - saturación (pureza, inverso a la cantidad de blanco).

Se pueden también visualizar los espacios de color en tres dimensiones, ver figura 1.16. El modelo RGB se puede ver como cubo, en donde los ejes corresponden a cada uno de los componentes primarios. El origen del cubo es el color negro y el vértice opuesto (el más lejano al origen) es el blanco. El modelo HSI se puede ver como un cilindro, donde la altura dentro del cilindro corresponde a la intensidad, la distancia al eje central a la saturación y el ángulo al croma.

Existe una forma directa de pasar la representación de color del modelo RGB al HSI y viceversa. Por ejemplo, las componentes en HSI se pueden calcular en base al RGB de la siguiente forma:

$$H = cos^{-1}\left(\frac{\frac{1}{2}(R-G)+(R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right) \tag{1.13}$$

$$S = 1 - \left(\frac{3min(R,G,B)}{R+G+B}\right) \tag{1.14}$$

$$I = \frac{1}{3}(R+G+B) \tag{1.15}$$

Estos modelos de color y otros, se verán a más detalle en el capítulo de color.

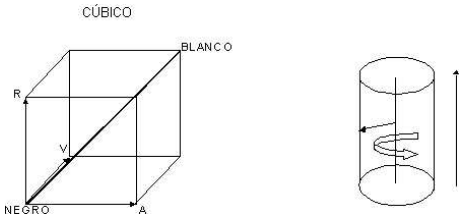


Figura 1.16: Representación gráfica de los espacios de color: (a) modelo RGB, (b) modelo HSI.

1.3 Digitalización de imágenes

Al muestrear la imagen para obtener una representación digital, hay dos factores importantes que considerar:

a) El intervalo de muestreo (resolución). b) El patrón espacial de los puntos de muestreo (*tesselation*).

1.3.1 Intervalo de muestreo

¿Qué tan próximas deben estar las muestras de la señal continua para que sea posible su reconstrucción? La respuesta nos la da el *teorema del muestreo* de Shannon [103]. Este dice que para lograr una recuperación completa, es necesario que la frecuencia de muestreo sea al menos dos veces mayor a la frecuencia mayor contenida en el espectro de la señal original. Esto se puede demostrar a partir de un análisis de Fourier del fenómeno de muestreo. Si no se cumple esto se presenta un fenómeno llamado "aliasing" en el cual las bajas frecuencias interfieren en las altas frecuencias, resultando en la pérdida de detalle de la imagen que se ve borrosa.

La figura 1.17 ilustra el muestro de una señal continua en el tiempo. El efecto de diferentes números de muestras o resolución para una imagen se puede observar en la figura 1.18.

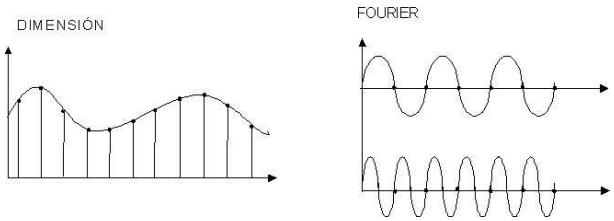


Figura 1.17: Muestreo de una señal continua.

1.3.2 Patrones espaciales

Si consideramos que los elementos de la imagen en realidad no son puntos sino celdas con un nivel de intensidad uniforme, entonces dichas celdas tienen cierta forma bidimensional. Existen tres tipos de arreglos de celdas (fig. 1.19):

- rectangular,
- triangular
- hexagonal.

Hay dos parámetros principales que considerar respecto a la forma de las celdas y que repercuten en diversos algoritmos de análisis de la imagen:

Figura 1.18: Simplificación de una imagen al tomar menos muestras: (a) imagen original, (b) resultado de promediar con máscara de 3x3, (c) resultado de promediar con máscara de 5x5, (d) resultado de promediar con máscara de 7x7.

1. Conectividad - determinar si ciertos elementos u objetos estan conectados o no. Para las celdas rectangulares se presentan problemas en este aspecto, ya que se puede definir en dos formas: 4 celdas u 8 celdas. En ciertos casos se presentan paradojas con ambas formas de definir vecindad, ver figura 1.19.

2. Distancia - determinar en forma consistente la distancia entre pixels. Para esto es conveniente que la distancia sea una métrica, que satisfaga lo siguiente:

   (a) $d(x, y) = 0 \leftrightarrow x = y$
   (b) $d(x, y) = d(y, x)$
   (c) $d(x, y) + d(y, z) \geq d(x, z)$

   Este aspecto es fácil de definir en un patron rectangular, pero es más complejo en los patrones triangulares y hexagonales.

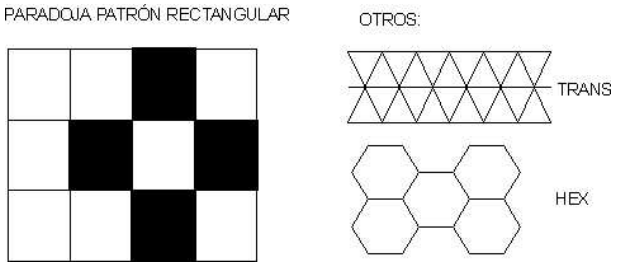


Figura 1.19: Patrones espaciales: (a) paradoja de conectividad con patrón rectangular, (b) patrones triangular y hexagonal.

1.4 Elementos de un Sistema de Visión

1.4.1 Dispositivos para visión

Existe diferentes dispositivos para la captura de imágenes. Dichas imágenes son digitalizadas y almacenadas en la memoria de la computadora. Una vez en la computadora, o en ocasiones desde el mismo dispositivo de captura, la imagen puede ser ya procesada.

Para la adquisición de la imagen se requiere de un dispostivo físico que sea sensible a una determinada banda del espectro electromagnético. El dispositivo produce una señal eléctrica proporcional al nivel de energía detectado, la cual es posteriormente digitalizada. Entre los dispositivos de captura o sensores se encuentran:

- cámaras fotográficas,
- cámaras de televisión (vidicón o de estado sólido - CCD)
- digitalizadores (scanners),
- sensores de rango (franjas de luz, laser),
- sensores de ultrasonido (sonares),
- rayos X,
- imágenes de tomografía,
- imágenes de resonancia magnética.

1.4.2 Arquitectura de un sistema de visión

Un sistema típico de visión por computadora, además de un dispositivo de captura, cuenta con al menos otros 4 elementos: un dispositivo de conversión de analógico a digital (A/D), una memoria de video, un elemento de procesamiento y un monitor. En la figura 1.20 se muestra la arquitectura básica de un sistema de visión. A continuación se describen los principales elementos:

- Dispositivo de captura. Dispositivo físico que es sensible a una determinada banda del espectro electromagnético. El dispositivo produce una señal eléctrica proporcional al nivel de energía detectado.
- Conversión A/D. Convierte la señal obtenida del dispositivo de captura en una señal digital.
- Memoria de video. Memoria semiconductora (RAM) en la que se almacena la imagen digitalizada. Normalmente la conversión A/D y la memoria de video se agrupan en un módulo conocido como *frame grabber* (captura de imágenes).
- Procesador. La memoria de video se acopla a un procesador de propósito general que permite operar sobre la imagen. Opcionalmente pueden existir otro procesador dedicado para captura y procesamiento de imágenes.
- Monitor. Generalmente se tiene un monitor que permita visualizar las imágenes adquiridas. El procesador y monitor pueden ser parte de una computadora de propósito general a la que se ha acoplado el *frame grabber*.

1.5 Niveles de análisis

Al considerar visión como un proceso de información, podemos analizarlo de diversas formas. Marr propone tres niveles:

1. Teoría computacional - El objetivo del proceso computacional, sus metas y las estrategias adecuadas para realizarlo (¿Qué?).

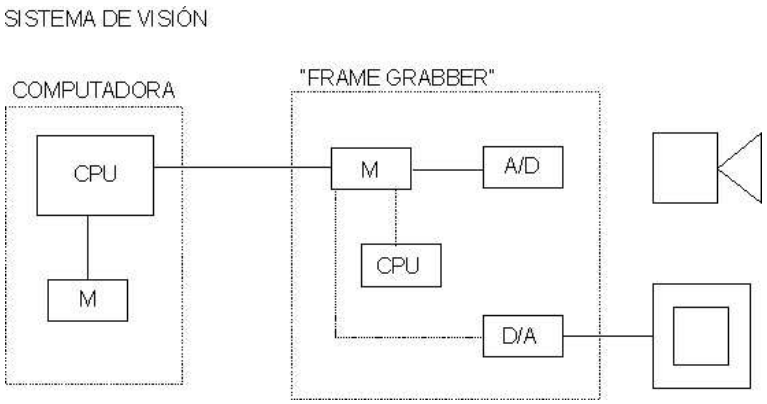


Figura 1.20: Arquitectura de un sistema de visión.

2. Representación y algoritmo - la descripción del proceso computacional, el representar las entradas y salidas, proponer el algoritmo para lograr dicha transformación (¿Como? - concepto).

3. Implementación - Como se realiza físicamente dicho proceso (¿Como? - físico).

El analizar un proceso a los diferentes niveles ayuda a su mejor entendimiento y realización.

1.6 Niveles de visión

Visión consiste en partir de una imagen (pixels) y llegar a una descripción (predicados, geometría, etc) adecuada de acuerdo a nuestro propósito. Como este proceso es muy complejo, se ha dividido en varias etapas o niveles de visión, en cada una se va refinando y reduciendo la cantidad de información hasta llegar a la descripción deseada. Se consideran generalmente tres niveles:

- Procesamiento de nivel bajo - se trabaja directamente con los pixels para extraer propiedades como orillas, gradiente, profundidad, textura, color, etc.
- Procesamiento de nivel intermedio - consiste generalmente en agrupar los elemento obtenidos en el nivel bajo, para obtener líneas, regiones, generalmente con el propósito de segmentación.
- Procesamiento de alto nivel - esta generalmente orientada al proceso de interpretación de los entes obtenidos en los niveles inferiores y se utilizan modelos y/o conocimiento *a priori* del dominio.

Aunque estas etapas son aparentemente secuenciales, esto no es necesario, y se consideran interacciones entre los diferentes niveles incluyendo retroalimentación de los niveles altos a los inferiores.

En los subsecuentes capítulos nos iremos adentrando en cada uno de los niveles de visión y en las técnicas que se utilizan para cada nivel.

1.7 Referencias

Existen varios libros que cubren los diferentes aspectos básicos de visión y procesamiento de imágenes. Entre los libros orientados a procesamiento de imágenes podemos mencionar a Gonzalez y Woods, *Digital Image Processing* (1992); Castleman, *Digital Image Processing* (1996); Parker,

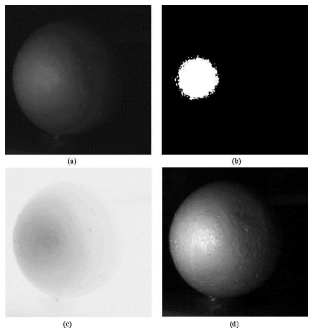
Algorithms for Image Processing and Computer Vision. En los libros orientados a visión, se encuentran, Ballard y Brown, *Computer Vision* (1982); Marr, *Vision* (1980); Pentland, *From Pixels to Predicates*; entre otros. Hay varias revistas dedicadas a temas de procesamiento de imágenes y visión, entre las que destacan: *International Journal of Computer Vision*, *CVGIP: Image Understanding*, *Image and Vision Computing*, *IEEE - Trans. on Pattern Analysis and Machine Intelligence*, *IEEE - Trans. on Systems, Man and Cybernetics.*

1.8 Problemas

1. ¿Qué es visión? ¿Qué es procesamiento de imágenes? ¿Cuál es la diferencia entre ambos?
2. Da dos ejemplos de problemas que se pueden resolver utilizando procesamiento de imágenes y dos que correspondan a visión.
3. Demuestra usando análisis de Fourier el teorema del muestreo.
4. Al digitalizar una imagen ¿qué tan "cerca" deben estar las muestras y porqué? ¿Qué pasa si no hay la suficiente resolución?
5. Considerando que cada pixel en una imagen se represente con 8 bits, y además se transmite un bit de inicio y uno de fin por "paquete" (pixel), cuántos segundos se requieren para transmitir una imagen de 1024 x 1024 pixels para una velocidad de transmisión de (a) 300 baud (bits por segundo), (b) 9600 baud, (c) 1 Mega baud.
6. Repite el problema anterior para imágenes a color, con 8 bits por banda, considerando que cada pixel es un paquete.
7. Define una métrica para distancia en arreglos de celdas rectangulares y hexagonales.
8. Analiza un proceso computacional de acuerdo a los niveles de análisis de Marr y describe cada uno de éstos.
9. Describe los tres principales niveles de visión. Especifíca las entradas y salidas a cada nivel, así como la información adicional que se requiera en cada uno.
10. Un proceso computacional lo podemos considerar desde tres puntos de vista: teoría computacional, algoritmo e implementación. Describe el proceso general de visión desde los tres aspectos.

1.9 Proyectos

1. Instala y prueba el "laboratorio de visión" en tu computadora. Prueba cargar y desplegar diferentes imágenes en diferentes formatos.



Capítulo 2

Mejoramiento de la imagen

2.1 Introducción

El objetivo de visión de bajo nivel o "procesamiento temprano" es hacer transformaciones directamente sobre la imagen para obtener información de las propiedades físicas de los objetos que estan en ella y que sean de mayor utilidad para los siguientes niveles de visión. Los principales atributos que se consideran importantes para obtener de una imagen son:

- discontinuidades u orillas,
- color,
- textura,
- gradiente y profundidad.

De tal forma, que podemos pensar que de la imágen original, se obtendra una "nueva imagen" por cada característica que se extraiga de la imagen -lo que Marr denomina el *Primal sketch* - llamadas "imágenes intrínsecas", como se ilustra en la figura 2.1.

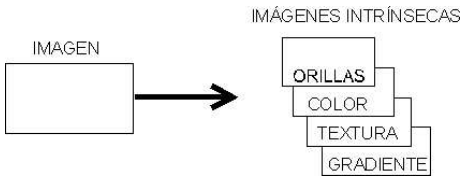


Figura 2.1: Imágenes instrínsecas o "*Primal Sketch*".

Previo a la obtención de estas carcterísticas es, muchas veces, necesario "mejorar" la imagen para resaltar aspectos deseados y eliminar los no deseados, tales como el ruido. Esta tárea tiene mucho en común con procesamiento de imágenes y, aunque es un campo muy amplio, nos concentraremos en tres tipos de técnicas que son frecuentemente utilizadas en la etapa de preprocesamiento:

- operaciones puntuales,
- filtrado,
- ecualización por histograma.

A continuación veremos en detalle cada una de estas técnicas.

2.2 Operaciones puntuales

Una *operación puntual* transforma una imagen de entrada a una imagen de salida de forma que cada pixel de la imagen de salida *sólo depende* del correspondiente pixel de la imagen de entrada; como se ilustra en la figura 2.2.

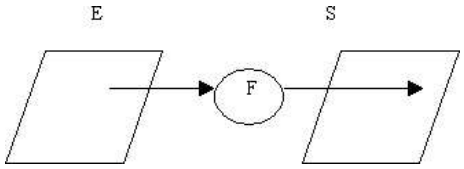


Figura 2.2: Operación puntual.

Una operación puntual se puede expresar matemáticamente como:

$$S[x, y] = f(E[x, y]) \tag{2.1}$$

Donde E es la imagen de entrada y S es la imagen de salida. La función f especifica el mapeo del nivel de gris de la entrada al nivel de gris de la salida. La forma en que se transforme la imagen depende de esta función. Esta función se puede interpretar gráficamente como se ilustra en la figura 2.3. La línea punteada a 45 grados en la figura indica la transformación en que cada pixel de salida es igual al de entrada (identidad).

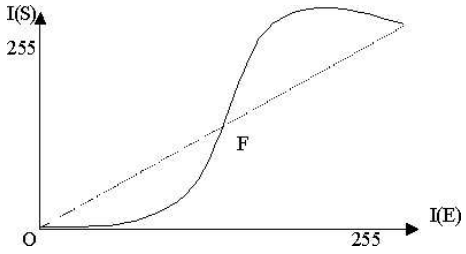


Figura 2.3: Función de transformación.

2.2.1 Binarización por umbral

La tarea de binarización, al menos en su forma básica, es una típica operación puntual. Para obtener una *imagen binaria* se hace una transformación no-lineal de la imagen de entrada, obteniéndose una imagen de salida en la cual cada pixel puede tomar alguno de dos valores: 0 y 1, negro y blanco, 0 y 255, etc. Para esto, se toma un valor de umbral T (*threshold*), de forma que:

$$S[x, y] = 1, E[x, y] > T \tag{2.2}$$

$$S[x, y] = 0, E[x, y] \leq T \tag{2.3}$$

La figura 2.4 muestra un ejemplo de una imagen que ha sido binarizada. Los pixeles con valores menores al umbral se muestran en negro (0) en caso contrario los pixeles se muestran en blanco (255).

Esta técnica se puede aplicar como una forma muy sencilla de "separar" un objeto de interés del resto de la imagen. Por ejemplo, el "objeto" de interés puede tomar el valor 1 y lo demás 0. El problema es como determinar el umbral. Por ejemplo, en la figura 2.4 no es posible determinar cual

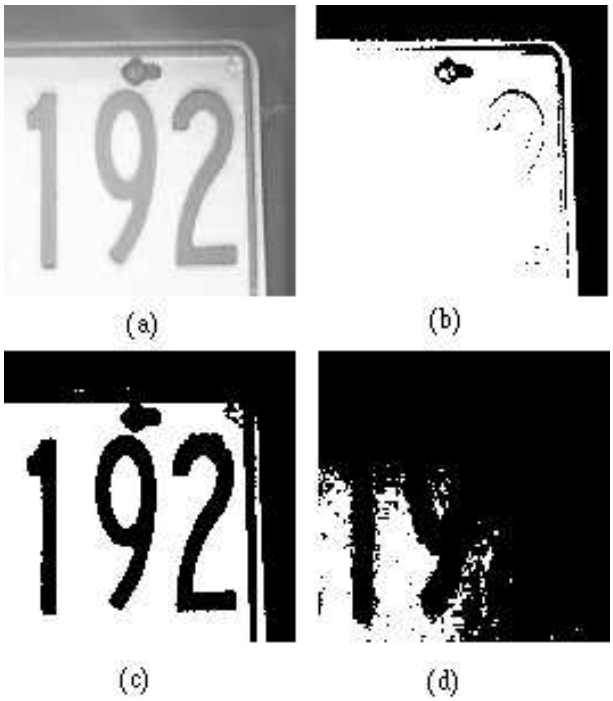


Figura 2.4: Ejemplo de binarización. (a) Imagen original. (b) Binarización con umbral en 150. (c) Binarización con umbral en 200. (d) Binarización con umbral en 250.

es el valor "óptimo" para separar los caracteres del fondo. En general esta técnica es de prueba y error, ya que el valor del umbral cambia entre las imágenes, así como para diferentes condiciones de iluminación y escenas a procesar. Una forma de determinar automáticamente este valor de umbral es utilizando su histograma de tonos de grises o segmentación por histograma, como se verá más adelante.

2.3 Transformaciones de intensidad

Una transformación de intensidad consiste en mapear los valores de intensidad de cada pixel a otros valores de acuerdo a cierta función de transformación. Las funciones de transformación pueden ser de dos tipos:

1. lineales,
2. no-lineales.

En las transformaciones lineales, se tiene una relación o función *lineal* de los valores de intensidad de los pixels de la imagen de salida respecto a la imagen de entrada. Los tipos de transformaciones lineales más comunmente utilizados son:

- Obtener el negativo de una imagen.
- Aumentar o dismunuir la intensidad (brillo de la imagen).
- Aumento de contraste.

Las funciones de transformación para cada uno de estos tipos se especifica gráficamente en la figura 2.5. Por ejemplo, para el negativo, el pixel de entrada (eje X) de intensidad 0 se transforma en un

pixel de salida (eje Y) de intensidad máxima, y el de entrada de intensidad máxima se tranforma en intensidad 0.

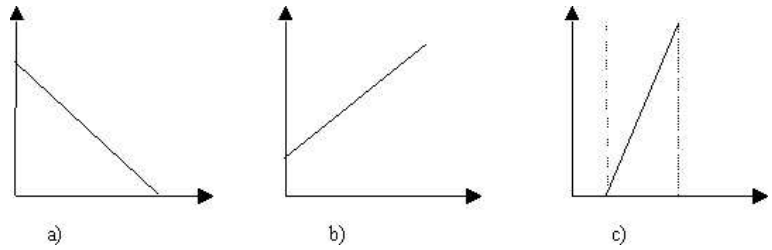


Figura 2.5: Transformaciones lineales. (a) Negativo. (b) Aumento de intensidad. (c) Aumento de contraste.

Las transformaciones no-lineales normalmente son funciones monotónicas de forma que mantienen la estructura básica de la imagen. Algunos ejemplos de transformaciones no-lineales son los siguientes:

- Expansión (o aumento) de contraste. Se incrementa el contraste, en forma diferente para distintos rangos de intensidades.
- Compresión de rango dinámico. Se reduce el rango de niveles de gris o intensidades de la imagen.
- Intensificación de un rango de niveles. Se aumenta la intensidad de un rango de niveles de gris de la imagen.

Estas transformaciones se muestran también en forma gráfica en la figura 2.6.

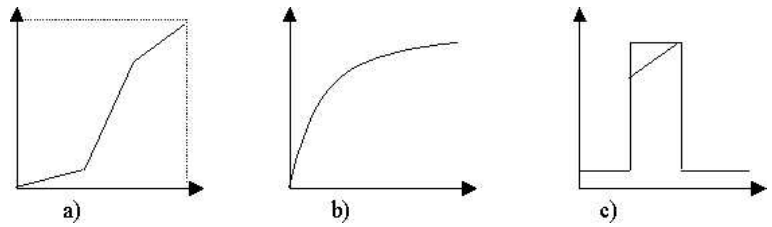


Figura 2.6: Transformaciones no lineales. (a) Expansión de contraste. (b) Compresión de rango dinámico. (c) Intensificación de un rango de niveles.

2.3.1 Aumento lineal del contraste

Utilizando el valor de intensidad mínimo y máximo en una imágen, podemos aumentar su contraste. La idea básica es llevar el valor mínimo (min) a cero y el máximo (max) a 255, pensando en imágenes monocromáticas (0-255). Esta transformación genera que las intensidades se espacíen de acuerdo a cierto factor o pendiente; el factor para este aumento *lineal* de contraste es:

$$C(x,y) = \left(\frac{I(x,y) - min}{max - min} * 255 \right) \tag{2.4}$$

Donde $I(x,y)$ es la imagen a procesar y $C(x,y)$ es la imagen con aumento lineal del contraste. Se puede verificar fácimente que para $I(x,y)$ en min, $C(x,y)$ resulta cero (el numerador es cero); para $I(x,y)$ en max, $C(x,y)$ resulta en 255 (cociente 1).

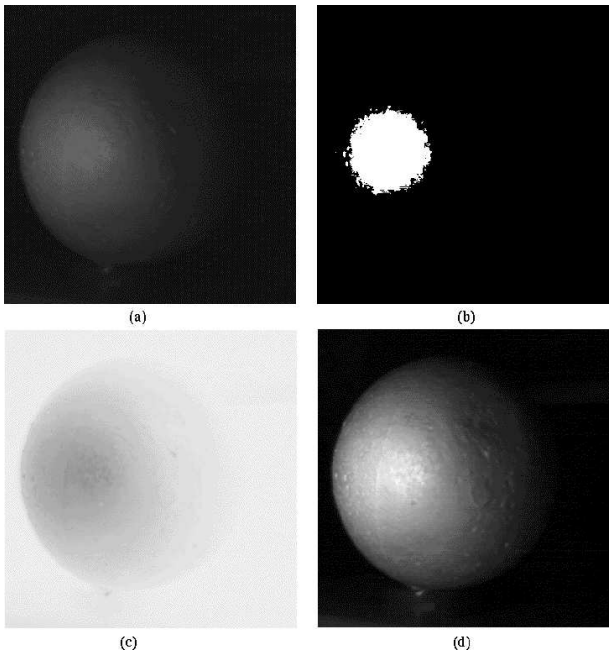


Figura 2.7: Ejemplo de operaciones puntuales: (a) imagen original. (b) binarización con umbral de 80. (c) negativo de la imagen original. (d) aumento lineal de contraste.

En la figura 2.7 se ilustra el resultado de aplicar diferentes operaciones puntuales a una imagen: binarización, negativo y aumento de contraste lineal.

Otra forma de hacer una expansión del contraste es utilizando el histograma de una imagen, mediante *ecualización por histograma*, lo cual veremos a continuación.

2.3.2 Ecualización del histograma

En esta sección se presentan los fundamentos matemáticos para realizar una ecualización por histograma, así como el procedimiento para su implementación. Para ello, antes veremos lo que es un histograma de una imagen.

Histograma de intensidades

Un histograma de una imagen es la distribución de cada nivel de intensidad dentro de la imagen, es decir nos da un estimado de la probabilidad de ocurrencia de cada nivel de gris (r).

$$p(r_k) = n_k/n \tag{2.5}$$

Donde $p(r_k)$ es la probabilidad del nivel k, n_k es el número de pixels que toma este valor y n es el número total de pixels en la imagen. En la figura 2.8 se muestra en forma gráfica el histograma de dos imágenes, una con amplio rango de intensidades y otra con un rango reducido.

El histograma nos presenta una descripción global de la imagen y sobre todo nos da una indicación del contraste en la imagen. De aquí que si modificamos el histograma, podemos controlar el contraste en la imagen.

Primero asumimos que el nivel de gris de la imagen, r, es una función continua y normalizada (entre 0 y 1). Deseamos realizar una transformación de forma que a cada nivel de gris r corresponda

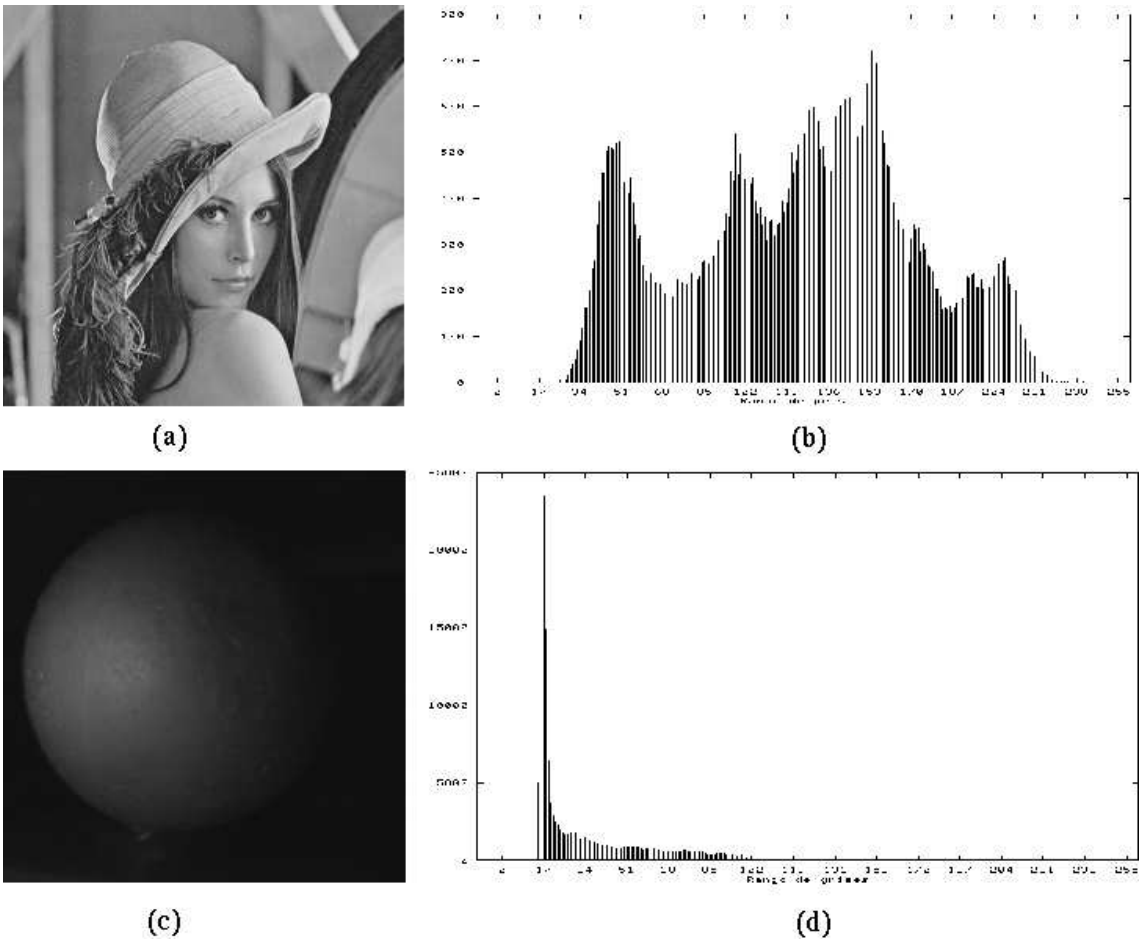


Figura 2.8: Ejemplos de histogramas: (a) Imagen con varias intensidades. (b) Su histograma mostrando un rango amplio de grises o alto contraste. (c) Imagen *obscura*. (d) Su histograma presenta un rango de grises reducido, es decir menor contraste.

un nuevo nivel s:

$$s = T(r) \tag{2.6}$$

Esta transformación debe satisfacer lo siguiente (ver fig. 2.9):

- T es una función monotónicamente creciente (mantener el orden).
- $0 \leq T \leq 1$ (mantener el rango).

Podemos considerar las distribuciones de $p(r)$ y $p(s)$ como densidades de probabilidad. Entonces de teoría de probabilidad:

$$p(s) = [p(r)dr/ds] \tag{2.7}$$

Si utilizamos como función de transformación la distribución acumulativa de r:

$$s = T(r) = \int p(r)dr \tag{2.8}$$

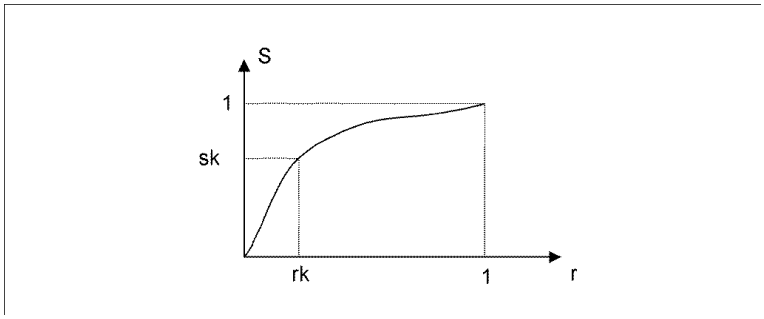


Figura 2.9: Función de transformación.

Entonces, derivando s respecto a r en la ecuación 2.8, obtenemos:

$$ds/dr = p(r) \tag{2.9}$$

Y, substituyendo 2.9 en la ecuación 2.7, finalmente llegamos a que:

$$p(s) = 1 \tag{2.10}$$

De forma que con esta transformación obtenemos una distribución uniforme para el histograma, maximizando así el contraste en la imagen.

En el caso discreto, la transformación se convierte en:

$$s(k) = T(r) = \sum_{i=0}^{k} n_i/n \tag{2.11}$$

Para $k = 0, 1, ..., N$, donde N es el número de niveles. Esto considera que ambos r y s están normalizados entre cero y uno. Para poner la imagen de salida en otro rango hay que multiplicar por una constante (p. ej., 255). Un ejemplo de aplicar esta técnica a una imagen de bajo contraste se presenta en la figura 2.10.

Esto se puede generalizar para obtener una distribución específica que no sea uniforme. También se puede aplicar en forma local a la imagen por regiones. Esta técnica provee en general muy buenos resultados para mejorar el contraste de una imagen.

2.4 Filtrado

El filtrar una imagen (f) consisten en aplicar una transformación (T) para obtener una nueva imagen (g) de forma que ciertas características son acentuadas o disminuidas:

$$g(x, y) = T[f(x, y)] \tag{2.12}$$

Podemos considerar que la señal (imagen) pasa a través de una caja o sistema (filtro) cuya salida es la imagen filtrada (ver fig. 2.11).

De acuerdo a la teoría de sistemas, al pasar una señal por un sistema lineal, la salida es la convolución de la transformación del sistema (función de transferencia) con la señal de entrada:

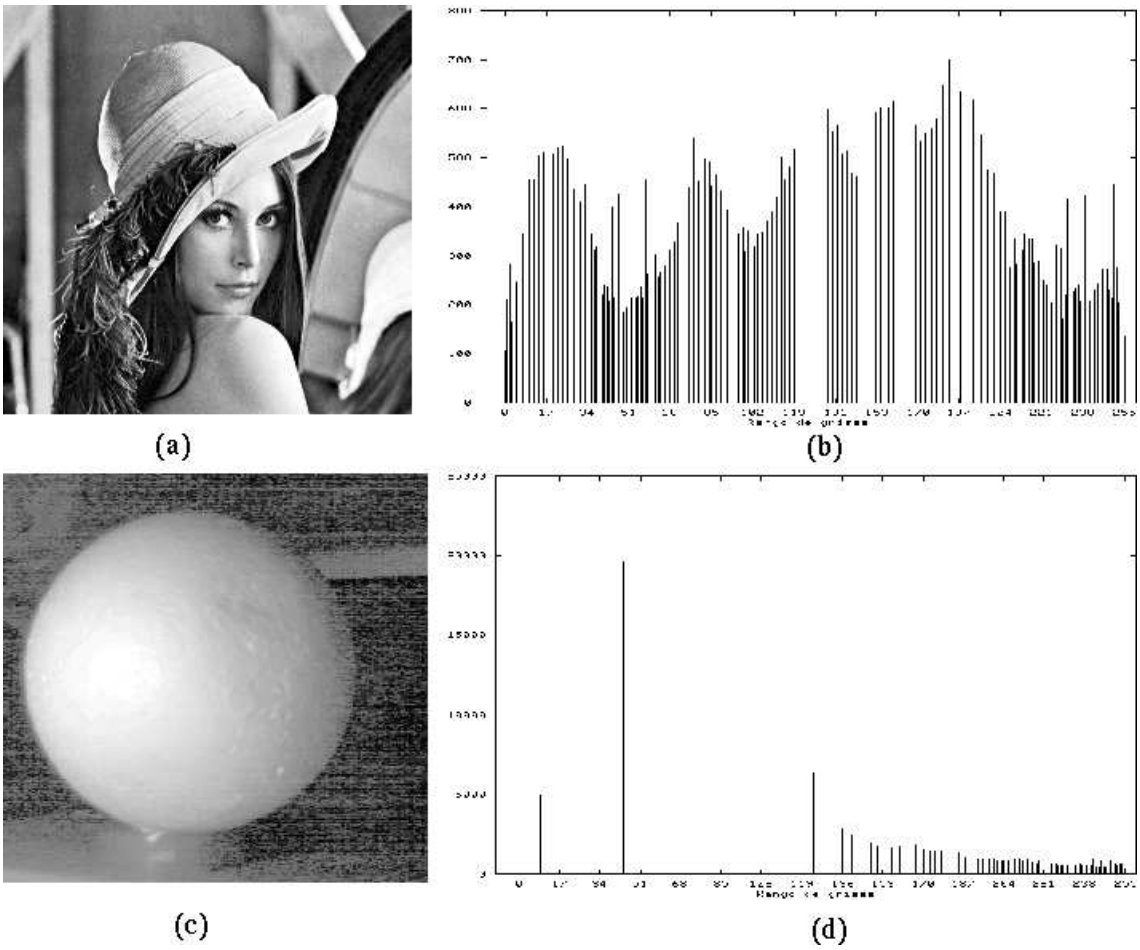


Figura 2.10: Ecualización por histograma. Comparese con la figura 2.8: (a) imagen ecualizada. (b) Histograma modificado. (c) Imagen ecualizada. (d) Histograma modificado.

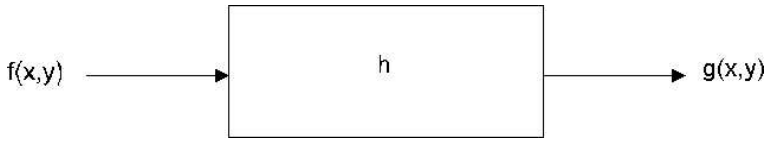


Figura 2.11: Proceso de filtrado.

$$g(x,y) = h(x,y) * f(x,y) \tag{2.13}$$

Por el teorema de la convolución, ésto corresponde a la multiplicación en el dominio de la frecuencia:

$$G(u,v) = H(u,v)F(u,v) \tag{2.14}$$

Por esto, podemos pensar en dos formas básicas de filtrar una imagen, realizarlo en el dominio espacial -que implica una convolución-, o en el dominio de la frecuencia -que implica sólo multiplicación pero dos transformaciones de Fourier (de espacio a frecuencia y viceversa). Ambos tipo de filtros han sido ampliamente estudiados y a continuación veremos sólo una introducción general y su aplicación en imágenes.

2.5 Filtrado en el dominio espacial

Las técnicas o filtros en el dominio espacial operan directamente sobre los pixels de la imagen. Operan en la vecindad de los pixels, generalmente mediante una *máscara* cuadrada o rectangular. Una máscara es una "pequeña" imagen que consiste de una serie de valores predeterminados para cada posición. La figura 2.12 ilustra un ejemplo de una máscara de 3 x 3, mas adelante veremos la función que realiza esta máscara sobre una imagen. La máscara se *centra* sobre el pixel de interés de forma que el nuevo valor del pixel depende de los pixels que cubre la máscara. En la figura 2.13 se ilustra en forma gráfica el proceso de filtrado o convolución con la máscara.

$w_{1,1}$	$w_{1,2}$	$w_{1,3}$
$w_{2,1}$	$w_{2,2}$	$w_{2,3}$
$w_{3,1}$	$w_{3,2}$	$w_{3,3}$

Figura 2.12: Ejemplo de máscara de 3x3

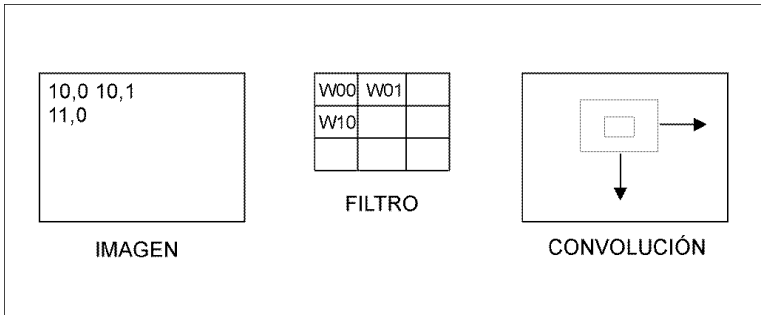


Figura 2.13: Filtrado en el dominio espacial.

A cada celda de la máscara le corresponde un peso o coeficiente (w), de forma que el nuevo valor del pixel es la sumatoria de el producto de los pixels vecinos con el peso correspondiente:

$$g(x,y) = \sum_i \sum_j f(i,j)w(i,j) \tag{2.15}$$

Generalmente, dividiendo sobre cierto valor para normalizar. Dicha máscara se aplica a cada pixel de la imagen, de forma que se realiza una convolución entre la máscara y la imagen original. El tamaño y los valores de los coeficientes determinarán el tipo de filtrado que se realize.

Las operaciones puntuales que se vieron en la sección anterior se pueden considerar como un filtro en el que el tamaño de la máscara es uno, es decir que el valor sólo depende de el pixel correspondiente. Otros tipos de filtros espaciales son los filtros de suavizamiento o pasa–bajo y los filtros de acentuamiento o pasa–alto, que analizaremos a continuación.

2.5.1 Filtros de suavizamiento

El objetivo de los filtros de suavizamiento es eliminar ruido o detalles pequeños que no sean de interés. Esto corresponde a un filtro pasa-bajos en el dominio de la frecuencia, es decir que se eliminan o reducen las altas frecuencias. En la figura 2.14 se muestra la respuesta de un filtro pasa-bajo en frecuencia (en una dimensión) y la correspondiente respuesta que debe tener en el dominio espacial.

Existen varios tipos de filtros para suavizamiento, los más comunes son:

- Promedio o media aritmética: Obtiene el promedio de los pixels vecinos ($w = 1$); es decir, todos los valores de la máscara son 1.

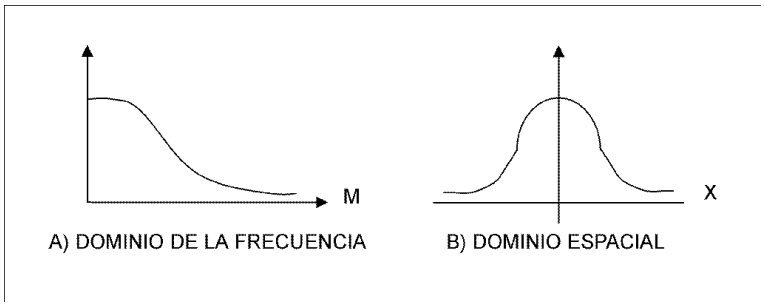


Figura 2.14: Filtro pasa-bajos: (a) en frecuencia, (b) en el dominio espacial.

- Mediana: Substituye el valor del pixel central por el de la mediana de los valores contenidos en el vecindario.
- Gaussiano: Aproximación a una distribución gaussina en dos dimensiones.

Considerando una media igual a cero, la función de transformación de un filtro tipo gaussiano es:

$$T(x, y) = e^{-[(x^2+y^2)/2\pi\sigma^2]} \tag{2.16}$$

Donde σ es la desviación estandar. Para un máscara de 3x3 los valores de un filtro gaussiano "típico" se muestran en la figura 2.15. La cantidad de "suavizamiento" que realiza el filtro gaussiano se puede controlar variando la desviación estandar y el tama no de la máscara.

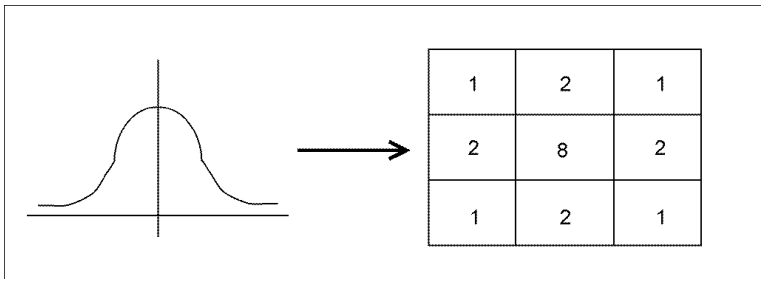


Figura 2.15: Máscara para filtro gaussiano de 3x3.

El filtro Gaussiano, en general, da mejores resultados que un simple promedio o media y se argumenta que la vista humana hace un filtrado de este tipo. El filtro Gaussiano "normal" o isotrópico tiene la desventaja de suavizar las orillas o discontinuidades, generando que se emborronen. Lo anterior genera problemas en las posteriores etapas de visión. El algoritmo de mediana es particularmente efectivo en imágenes con poco ruido. Su efectividad decrece drásticamente en imágenes ruidosas.

La figura 2.16 ilustra el resultado de aplicar a una imagen diferentes tipos de filtros pasa-bajo.

2.5.2 Filtros de acentuamiento

El objetivo de los filtros de acentuamiento es intensificar los detalles y cambios bruscos de intensidad mientras atenua las bajas frecuencias. El resultado es un *acentuamiento* de las orillas (*edge sharpening*). Se conocen como filtros de pasa-alto porque dejan pasar las altas frecuencias y eliminan las bajas frecuencias, en forma inversa al filtro pasa-bajo. En la figura 2.17 se muestra como se reducen las bajas frecuencias y se mantienen las altas.

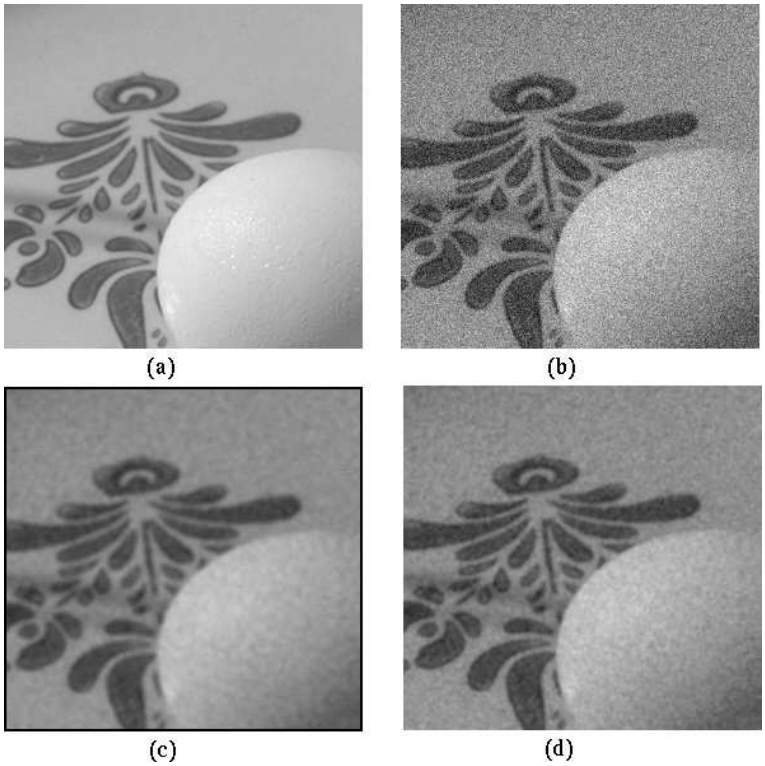


Figura 2.16: Filtros pasa-bajo en el dominio espacial. (a) imagen original, (b) imagen corrupta con ruido gaussiano. (c) resultado de aplicar un filtro promedio con máscara de 5x5. (d) resultado de filtro gaussiano, σ=1.0.

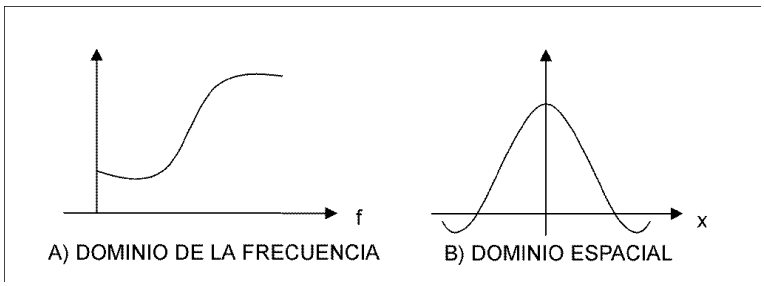


Figura 2.17: Filtro pasa-alto: (a) en frecuencia, (b) en el dominio espacial

Existen también varias formas de implementar este tipo de filtrado. Una forma típica de un filtro pasa-altos es una máscara del tipo de la figura 2.18. Para este filtro la suma de los pesos es cero, de forma que una región de intensidad constante resultaría en un valor 0. Nótese que a diferencia del filtro de suavizamiento los pesos de los vecinos son negativos, este efecto substractivo genera la acentuación de los cambios de intensidad.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 2.18: Máscara de 3x3 para un filtro pasa-alto simple.

Otra forma de implementar este tipo de filtrado es restando a la imagen original el resultado de un filtro pasa-bajo:

$$PA = original - PB \tag{2.17}$$

Donde PA representa la imagen resultante de aplicar un filtro pasa-alto y PB de un filtro pasa-bajos a la imagen "original".

2.5.3 Filtro para énfasis de altas frecuencias

El filtrado de acentuamiento o pasa altos presenta sólo las discontinuidades, atenuando fuertemente las bajas frecuencias y haciendo que "desaparezcan" las regiones homogéneas. Un tipo de filtro que aún acentuando las altas frecuencias preserva las bajas es el filtro "énfasis de altas frecuencias" (*high boost*).. Para obtener una imagen con énfasis de altas frecuencias (EA), se puede considerar que se multiplica la imagen original por una constante A, esta constante debe ser mayor que uno para que acentúe.

$$EA = (A)original - PB \tag{2.18}$$

Eso es equivalente a la siguiente expresión:

$$EA = (A-1)original + PA \tag{2.19}$$

En la práctica no es necesario hacer exactamente esta operación, sino se implementa haciendo la celda central del filtro pasa-alto:

$$w = 9A - 1 \tag{2.20}$$

Como se ilustra en la figura 2.19.

-1	-1	-1
-1	$9A-1$	-1
-1	-1	-1

Figura 2.19: Máscara de 3x3 para un filtro pasa-alto con énfasis en las altas frecuencias.

En la figura 2.20 se muestra el resultado de aplicar a una imagen diferentes tipos de filtros pasa-alto.

2.6 Filtrado en el dominio de la frecuencia

En el caso de filtrado en el dominio de la frecuencia se hace una transformación de la imagen utilizando la transformada de Fourier. Entonces los filtros se aplican a la función (imagen) transformada y, si es necesario, se regresa al dominio espacial mediante la transformada inversa de Fourier. Para esto veremos primero un repaso de la transformada de Fourier.

2.6.1 Transformada de Fourier

Dada una función $f(x)$ de una variable real x, la transformada de Fourier se define por la siguiente ecuación:

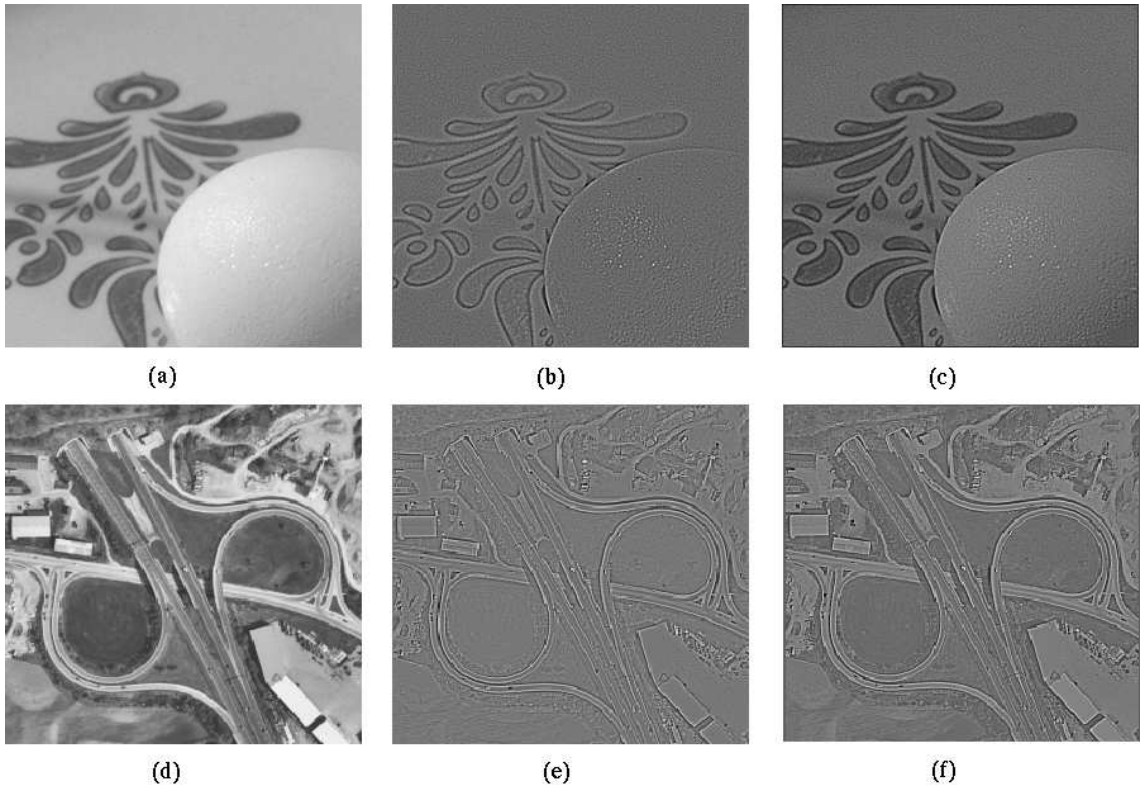


Figura 2.20: Filtros pasa-alto en el dominio espacial: (a) y (d) imágenes originales, (b) y (e) resultado de filtro pasa-alto simple, (c) y (f) resultado de filtro de énfasis de altas frecuencias. Factor: $A = 1.1$.

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{[-j2\pi ux]} dx \tag{2.21}$$

Donde $j = \sqrt{-1}$.

Dada $F(u)$ se puede obtener $f(x)$ mediante la transformada inversa de Fourier:

$$f(x, y) = \int_{-\infty}^{\infty} F(u) e^{[j2\pi ux]} du \tag{2.22}$$

Las ecuaciones anteriores constituyen lo que se conoce como el par de transformación de Fourier.

En general F es compleja, y la podemos descomponer en su magnitud y fase:

$$F(u) = R(u) + jI(u) = |F(u)| e^{j\Phi(u)} \tag{2.23}$$

En el caso de una función de dos dimensiones, $f(x, y)$, como es el caso de una imagen, el par de transformación de Fourier es el siguiente:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{[-j2\pi(ux+vy)]} dxdy \tag{2.24}$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{[j2\pi(ux+vy)]} dudv \tag{2.25}$$

Si consideramos una imagen digital, entonces se requiere lo que se conoce como la transformada discreta de Fourier. Para esto se supone que se ha discretizado la función $f(x)$ tomando N muestras separadas Δx unidades. Entonces la transformada discreta de Fourier se define como:

$$F(u) = (1/N) \sum_{x=0}^{N-1} f(x) e^{[-j2\pi ux/N]} \tag{2.26}$$

Para $u = 1, 2, ..., N-1$. La transformada inversa es:

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{[j2\pi ux/N]} \tag{2.27}$$

Para $x = 1, 2, ..., N-1$.

En el caso de dos dimensiones se tienen las siguientes expresiones:

$$F(u,v) = \left(\frac{1}{MN}\right) \sum\sum f(x,y) e^{[-j2\pi(ux/M+vy/N)]} \tag{2.28}$$

$$f(x,y) = \sum\sum F(u,v) e^{[j2\pi(ux/M+vy/N)]} \tag{2.29}$$

Algunas propiedades de la transformada de Fourier importantes para visión son las siguientes:

- *Separabilidad:* Se puede separar la transformada en cada dimensión, de forma que se puede calcular en renglones y luego columnas de la imagen.
- *Traslación:* Multiplicación por un exponencial corresponde a traslación en frecuencia (y viceversa). Se hace uso de esta propiedad para desplazar F al centro de la imagen:

$$e^{[j2\pi(Nx/2+Ny/2)/N]} = e^{[j2\pi(x+y)]} = (-1)^{(x+y)} \tag{2.30}$$

- *Rotación:* Rotando f por un ángulo se produce el mismo rotamiento en F (y viceversa).
- *Periodicidad y simetría:* La transformada de Fourier y su inversa son simétricas respecto al origen y periódica con un periodo $= N$.
- *Convolución:* Convolución en el dominio espacial corresponde a multiplicación en el dominio espacial (y viceversa).

En la figura 2.21 se ilustran en forma gráfica algunas de las propiedades de la transformada de Fourier.

2.6.2 Filtrado en frecuencia

El filtrado en el dominio de la frecuencia consiste en obtener la transformada de Fourier, aplicar (multiplicando) el filtro deseado, y calcular la transforma da inversa para regresar al dominio espacial (ver figura 2.22).

Existen muchas clases de filtros que se pueden aplicar en el dominio de la frecuencia. Dos de los filtros más comunes son el llamado *filtro ideal* y el *filtro Butterworth*. Ambos tipos de filtros pueden ser pasa-altos y pasa-bajos.

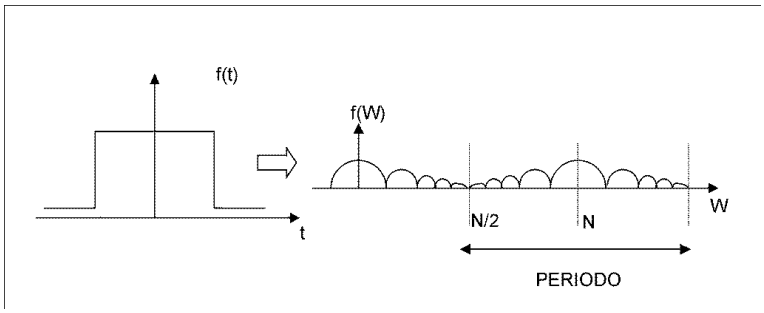


Figura 2.21: Algunas propiedades de la transformada de Fourier.

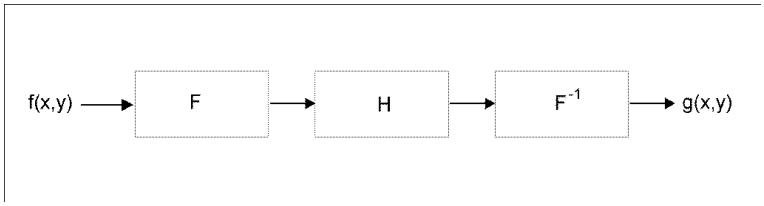


Figura 2.22: Filtrado en el dominio de la frecuencia.

El filtro ideal pasa–bajos tiene una función de transferencia $H(u, v)$ que es igual a 1 para todas las frecuencias menores a cierto valor (D_0) y cero para las demás frecuencias. Un filtro ideal pasa–altos tiene la función de transferencia opuesta, es decir es cero para todas las frecuencias menores a cierto valor y uno para las demás frecuencias. En la figura 2.23 se muestra la función de transferencia de un filtro ideal pasa–bajos.

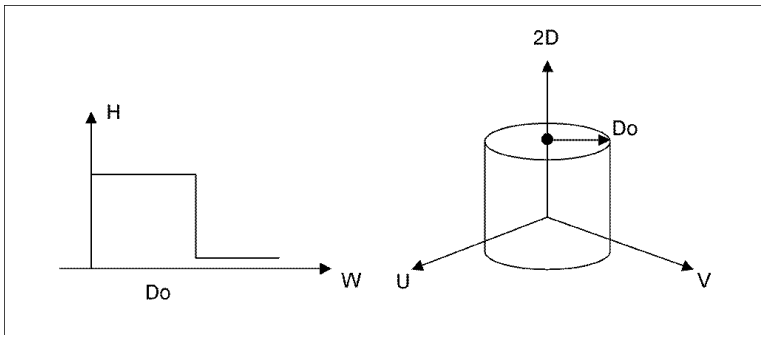


Figura 2.23: Función de transferencia de un filtro ideal pasa–bajos: (a) función en una dimensión (W), (b) función en dos dimensiones (U, V).

El filtro Butterworth tiene una función de transferencia más "suave" que generalmente da mejores resultados. Por ejemplo, la función de transferencia de un filtro Butterworth pasa-bajo de orden n y distancia D al origen se define como:

$$H(u, v) = \frac{1}{1 + \left(\frac{\sqrt{u^2+v^2}}{D_0}\right)^{2n}} \tag{2.31}$$

Esta función de transferencia se ilustra gráficamente en la figura 2.24.

Existe una manera más eficiente de hacer las transformada discreta de Fourier denominada transformada rápida de Fourier (FFT). De cualquier forma el procesamiento es generalmente más costoso y tienden a utilizarse más en la práctica los filtros en el dominio espacial. Sin embargo, se logra mayor precisión y flexibilidad en el dominio de la frecuencia y en ciertos casos vale la pena

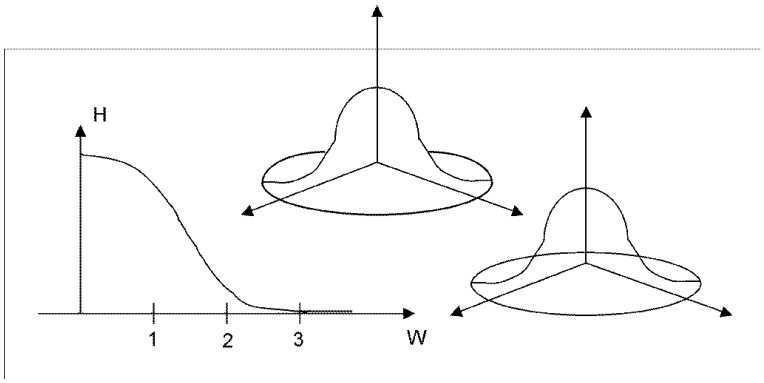


Figura 2.24: Función de transferencia de un filtro Butterworth pasa-bajo.

el costo computacional adicional.

2.7 Filtrado adaptable

Uno de los problemas al aplicar filtros pasa bajo o de suavizamiento para eliminar ruido, es que también se pueden eliminar atributos de la imagen que son importantes para la siguientes etapas de visión. Como veremos en el siguiente capítulo, las orillas o bordes en la imagen son muy importantes, y éstos tienden a "emborronarse" al aplicar un filtro de suavizamiento.

Una alternativa para al mismo tiempo remover ruido y preservar las orillas es mediante filtros selectivos o *adaptables*, que tratan de suavizar sólo en ciertas regiones de la imagen. La selección de donde suavizar se hace normalmente en función del gradiente local (como varía la imagen en una pequeña región), de forma que se filtre el ruido y no las orillas. A este tipo de filtros se les conoce como filtros no-lineales, que mantienen las orillas (*edgepreserving*) o adaptables.

El ejemplo más sencillo de esta clase de filtros es el filtro de mediana, que mencionamos en la sección 2.5.1. El filtro de mediana intenta preservar las orillas mientras que suaviza (promedia) regiones homogéneas. Aunque da mejores resultados que un filtro promedio, el filtro de mediana no logra resultados óptimos en el compromiso de preservar orillas y eliminar ruido. Por ello se han desarrollado otras técnicas más sofisticadas entre las que destacan:

- difusión anisotrópica,
- campos aleatorios de Markov,
- filtrado gaussiano no–lineal,
- filtrado gaussiano adaptable.

Veremos el filtrado gaussiano adaptable a continuación, para mayor información de las demás técnicas consultar la sección de referencias al final del capítulo.

2.7.1 Filtrado gaussiano adaptable

La idea del filtrado gaussiano adaptable es aplicar filtros gaussianos a la imagen variando la desviación estandar del filtro (σ) en función del gradiente local de cada región de la imagen. Para estimar el gradiente en diferentes regiones de la imagen se utiliza el concepto de *escala local*.

La escala se refiere al nivel de detalle que se tiene en una imagen; es decir, a *escalas grandes* podemos observar todos los detalles de los objetos, y al ir reduciendo la escala se va perdiendo

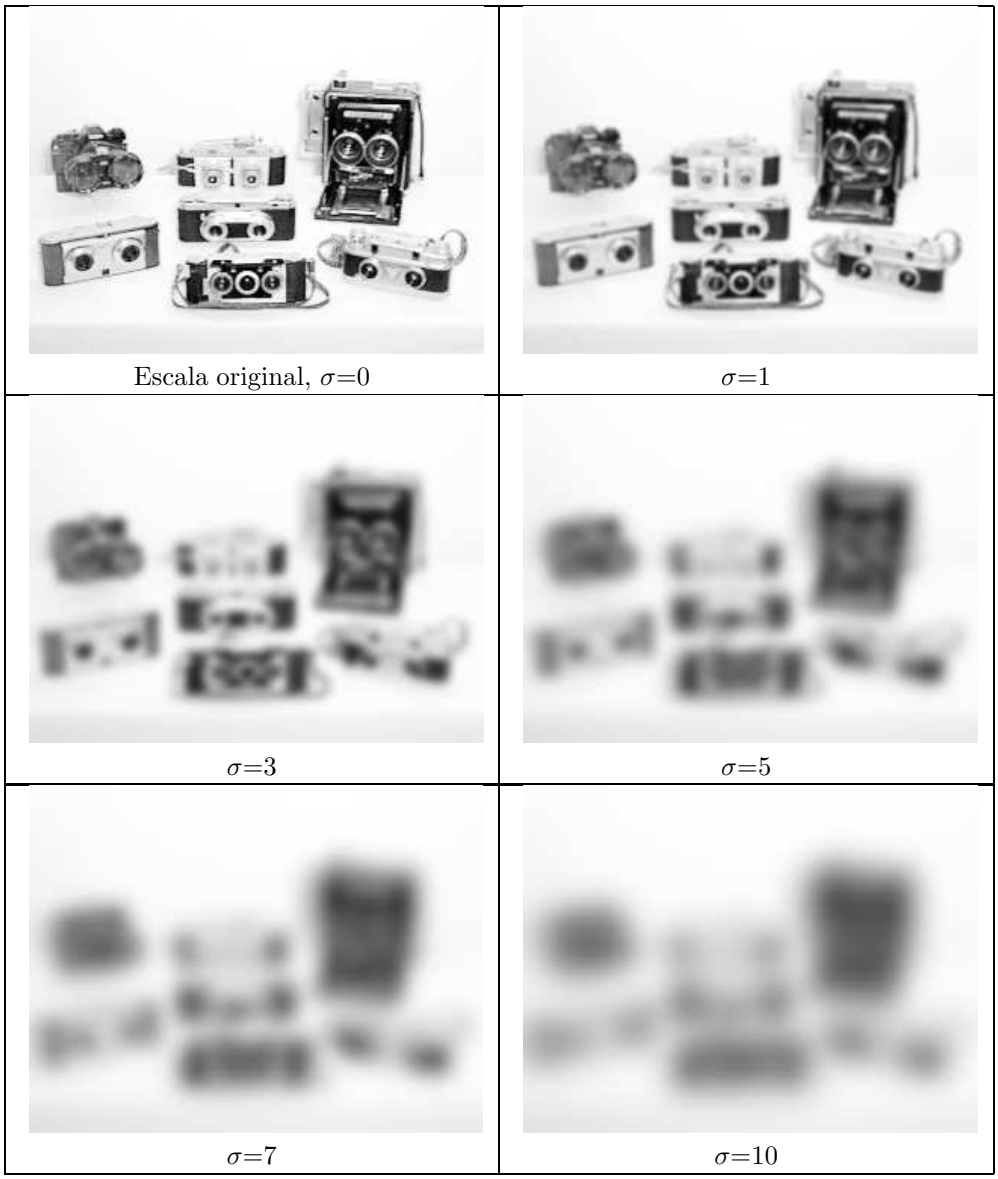


Figura 2.25: Imágenes variando la escala (σ).

información (como si fuera una imagen borrosa). Una forma de ilustrar la escala es mediante el filtrado de la imagen con filtros gaussianos de diferentes σ, que al ir aumentando va reduciendo la escala. La figura 2.25 muestra una imagen a diferentes escalas variando la σ.

Si se desea obtener cierta información de una imagen, hay una escala óptima para ello, en la cual se tiene el nivel de detalle necesario, pero no más. Por ello, se puede considerar que existe una escala local óptima de cada parte de la imagen. Dicha escala se puede obtener mediante un compromiso entre el minimizar el número de bits para representar la región (menor resolución) y a la vez minimizar el error de esta aproximación; utilizando el principio de longitud de descripción mínima (MDL).

Al filtrar una imagen (o sección de una imagen) con un filtro gaussiano, podemos considerar que la imagen filtrada aproxima la original, mas un cierto error:

$$I(x,y) = I_\sigma(x,y) + \epsilon(x,y) \tag{2.32}$$

En base al principio MDL, la longitud de descripción de la imagen se puede obtener combinando la *longitud* de la imagen filtrada más la longitud del error. La longitud de la imagen filtrada es inversamente proporcional a la σ del filtro, ya que al ir suavizando más la imagen, se requieren menos bits para representarla. Se puede demostar (Gómez et al.) que la longitod total es equivalente a:

$$longI(x, y) = (\lambda/\sigma^2) + \epsilon^2 \tag{2.33}$$

Donde λ es una constante. Entonces, podemos obtener la longitud para diferentes valores de σ (dentro de un rango obtenido experimentalmente) y seleccionar, para cada pixel, el filtro que de la menor longitud. Este filtro sería el óptimo de acuerdo al principio MDL.

En base a lo anterior, se integra el siguiente algoritmo para filtrado gaussiano adaptable:

1. Seleccionar la escala local para cada región (pixel) de la imagen, obteniendo la σ óptima.
2. Filtrar cada región (pixel) con el filtro gaussiano con la σ óptima.
3. Obtener la imagen filtrada.

El resultado de aplicar diferentes tipos de filtros adaptables a una imagen se puede observar en la imagen 2.26.

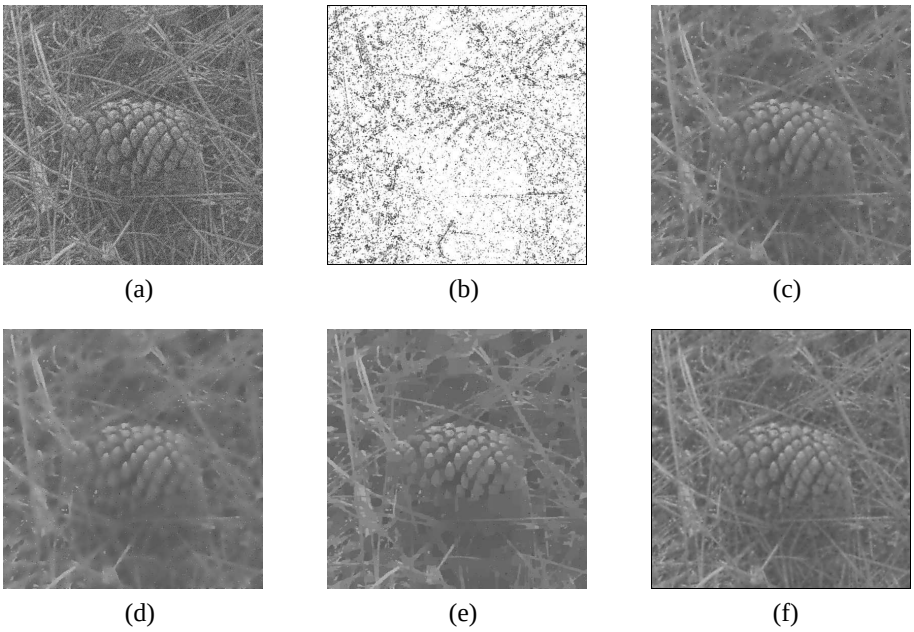


Figura 2.26: Ejemplo de filtrado gaussiano adaptable. (a) Imagen de un cono con ruido gaussiano. (b) Mapa de escalas locales. (c) Imagen filtrada con difusión anisotrópica, $k = 10$, después de 50 iteraciones; y (d) 80 iteraciones. (e) Filtrado gaussiano no-lineal. (f) Filtrado gaussiano adaptable.

2.8 Referencias

Para mayor información sobre las técnicas de mejoramiento de imágenes, consultar alguno de los libros especializados en procesamiento de imágenes como el de Gonzalez y Woods [28] o el de Castleman [11]. Un tratamiento más extensivo de los fundamentos de la transformada de Fourier se puede encontrar en Papoulis [85].

Una de las actuales áreas en el mejoramiento de imágenes o *image enhancement* es el de tratar la imágen a través de un banco de filtros Gaussianos. Este *espacio de escalas* [136] que se genera ha servido, desde mediados de los 80s, como base para técnicas de supresión de ruido. Ejemplos de estas técnicas son el "suavizamiento adaptable" [14, 99], la difusión isotrópica [64] y la difusión anisotrópica [89, 90]. Existen otros tipos de filtrado para mejorar disminuir el ruido (pasa bajas) y acentuar las discontinuidades principales. Por ejemplo, la técnica de difusión dirigida por tensores ("tensor valued diffusion") [134, 135] modifica el aspecto del kernel Gaussiano (formas elípticas; con esta forma realiza fitrados muy finos y no a través de las discontinuidades. El principal inconveniente de las anteriores técnicas ha sido el difícil ajuste de los parámetros involucrados. Aun cuando las anteriores técnicas son iterativas, se han desarrollado otros enfoques los cuales son técnicas directas, mas estables, que no necesitan ajustar más que un parámetro [27, 26, 20]. Esta área esta en contínuo movimiento y se recomienda al lector consultar las principales referencias especializadas del tema.

2.9 Problemas

1. Una forma de transformación es obtener una imagen por cada "bit" del valor del pixel en binario. Suponiendo cada pixel representado por un byte (8 bits), se tendrían 8 "planos" que representarían a la imagen a diferentes niveles de detalle. Definir la función de transformación para obtener estas imágenes de salida.

2. ¿Qué es ecualización por histograma? ¿Qué efecto tiene en la imagen?

3. ¿Cuál es la diferencia entre el filtrado en el dominio de la frecuencia y el filtrado en el dominio espacial? ¿Qué ventajas y desventajas tienen los dos enfoques?

4. Demuestra que si volvemos a ecualizar por histograma una imagen previamente ecualizada, el resultado es el mismo (no hay cambio).

5. ¿Qué objetivos tiene el filtrado que elimina altas frecuencias y el que las acentua? Da ejemplos de máscaras para ambos tipos de filtros.

6. Obten las máscaras para un filtro Gaussiano de 5×5 pixels, y d.s. $= 1$ y 3 pixels.

7. Considera una imagen de 8 x 8 con 8 niveles de gris, que tiene un fondo negro (0) y un cuadrado de 4 x 4 con nivel 4 al centro. Ilustra la aplicación de un filtro pasa-bajos (promedio) y pasa-altos a dicha imagen, obteniendo la nueva imagen.

8. Considera la siguiente imagen (binaria):

0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

   Da el resultado (como imagen) de aplicar un filtro de mediana a dicha imagen. Especifica que consideraste para el "borde" de la imagen.

9. Se quiere filtrar una imagen eliminando altas y bajas frecuencias, pero con un sólo filtro. (a) Diseña un filtro en el dominio espacial para hacer esto y da los valores para una máscara de 3 x 3. (b) Diseña un filtro similar en el dominio de la frecuencia y da su magnitud mediante una gráfica en 2-D.

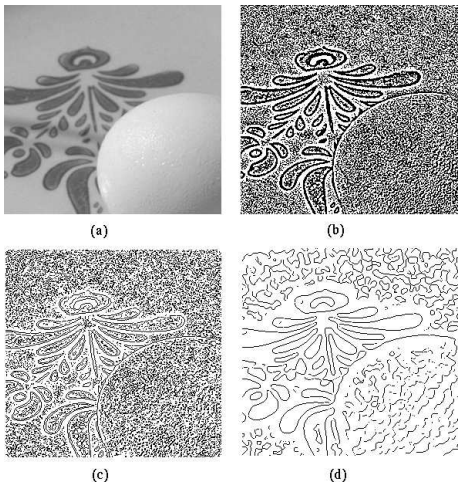
10. Comenta que pasa en el *límite* al aplicar repetidamente un filtro pasa bajos espacial de 3 x 3 a una imagen (puedes despreciar el efecto de las orillas de la imagen).

2.10 Proyectos

1. Implementar en el laboratorio las siguientes operaciones puntuales: (a) aumento de constraste lineal, (b) ecualización por histograma. Desplegar las imágenes resultantes, considerando que se requieren normalizar los valores de intensidad al rango original (0–255).

2. Implementar en el laboratorio los filtros espaciales básicos: (a) pasa bajos, (b) pasa altos; utilizando máscaras de 3 x 3. Desplegar las imágenes resultantes.

3. Implementar en el laboratorio un filtro con máscara cuadrada general (se puede variar tamaño y valores). Probar con varios filtros gaussianos de diferentes desviaciones (*sigmas*), aplicando a diferentes imagenes. Desplegar las imágenes resultantes.



Capítulo 3

Detección de orillas

3.1 Introducción

Diversos experimentos psicofisiológicos han mostrado que el sistema visual humano utiliza una amplia gama de fuentes de información, tales como las sombras, proporciones, longitudes, color, curvatura e intensidades. De las anteriores, las variaciones en intensidad u "orillas" se cuentan entre las más importantes. Aún si una imagen carece de información tridimensional, textura o sombras podemos reconocer el objeto utilizando sus orillas o silueta, ver figura 3.1.

Figura 3.1: Podemos reconocer un "dálmata" aún si la imagen carece de información tridimensional, sombras o textura.

La información de orillas es procesada por el sistema visual primario, en donde se encuentran células especializadas que responden a las discontinuidades. Las visión humana utiliza las orillas de manera jerárquica, agrupándolas y utilizando la experiencia visual hasta poder reconocer objetos más complicados que líneas, tales como rostros y objetos geométricos. Este subsistema de la visión biológica ocasionalmente "completa" bordes que están, al parecer, ocluídos o implícitos. Los *contornos subjetivos* de Kanizsa, figura 3.2, son un ejemplo donde el sistema visual "completa" bordes y modifica las intensidades, es decir, se completan con figuras regulares y aparecen "más" brillantes.

Detectar orillas es una tarea particularmente importante en visión por computadora. Los límites o bordes físicos, discretizados como variaciones de intensidad, son un punto de partida para tareas de bajo nivel como detección de esquinas, bordes y compresión de imagenes; y son la base de tareas de nivel intermedio como la separación o segmentación de los diferentes objetos en una imagen.

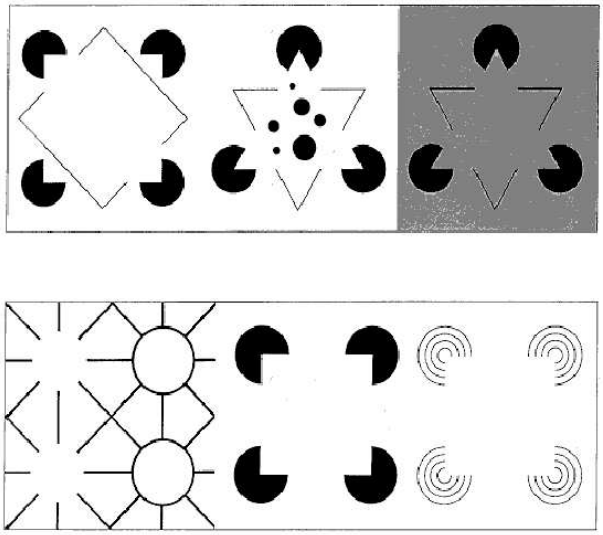


Figura 3.2: El sistema visual automáticamente "completa" las figuras agregando las orillas faltantes, como en los *contornos subjetivos* de Kanizsa.

La manera más común para detectar orillas es utilizar algún tipo de derivada o diferencial, aplicado normalmente en un vecindario "pequeño". La derivada nos permite calcular las variaciones entre un punto y su vecindario. Viendo la imagen como una función, un contorno implica una descontinuidad en dicha función, es decir donde la función tiene un valor de gradiente o derivada "alta" (ver figura 3.3).

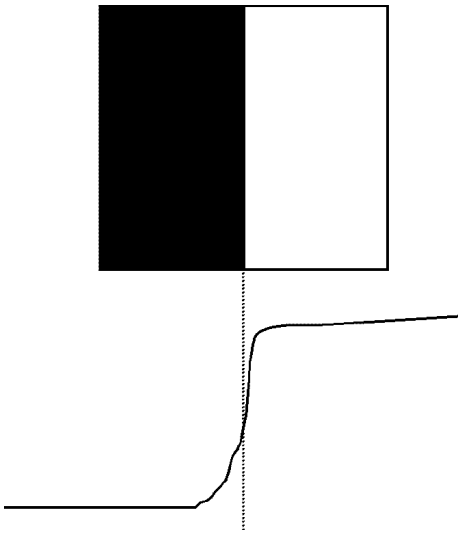


Figura 3.3: Ejemplo de discontinuidades. Arriba se muestra una imagen con una discontinuidad en intenisdad entre la parte izquierda y derecha. En la figura de abajo se grafica la intensidad de un "corte" horizontal de la imagen (un renglón) en el que se observa el alto gradiente en la parte correspondiente a la descontinuidad.

Al apreciar detenidamente un borde en una imagen vemos que éste se encuentra integrado de "orillas locales" u orillas individuales. En visión por computadora cada una de estas orillas locales (figura 3.4) son integradas o unidas, en etapas posteriores, en algo más útil que pixeles aislados, a estos les llamaremos *bordes*.

La detección de orillas, como veremos más adelante, es bastante sensible al ruido lo cual dificulta el proceso de integración de bordes. Debido a esta dificultad han surgido una gran cantidad

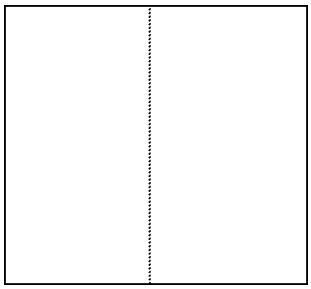


Figura 3.4: Orillas locales. Si puede ver el borde o discontinuidad de la imagen de la fig. 3.3 como constituido por una serie de "puntos" que corresponden a orillas locales.

de técnicas de detección de orillas y es, quizá, el tema con mayor número de artículos publicados en la literatura especializada en visión. El principal problema a lo que se enfrentan cada uno de estos trabajos es el como reconocer las orillas "visualmente relevantes", que pertenecen a contornos de interés, para diferenciarlas de otras orillas "falsas" generadas por fenómenos como ruido, sombreado, textura, etc.

Después de obtener las orillas, es común que se seleccionen de las orillas "relevantes", utilizando cierta información del contexto o del dominio. Tales técnicas "forzan" a detectar círculos, líneas, objetos largos, cambios "suaves", etc. Este postprocesamiento se conoce como *task-driven* o dependiente de la tarea a realizar.

Las técnicas de detección de orillas se pueden clasificar en:

- operadores de gradiente,
- múltiples respuestas a diferentes orientaciones,

en tanto que los post–procesamientos para crear bordes se pueden clasificar en:

- relajación,
- seguimiento de orillas.

En las siguientes secciones analizaremos cada uno de ellos.

3.2 Operadores de gradiente

Las técnicas clásicas de detección de orillas se basan en diferenciar a la imagen, esto es, encontrar la derivada respecto a los ejes x y y, o gradiente. El gradiente de una función $f(x, y)$ se define como:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \tag{3.1}$$

La magnitud[1] del gradiente (∇f) se calcula como:

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2} \tag{3.2}$$

[1]En la práctica puede ser conveniente evitar el cálculo de la raíz cuadrada y utilizar los valores absolutos de las diferencias.

En el caso discreto, podemos aproximar la derivada tomando simplemente la diferencia entre dos valores contiguos. Si consideramos una sección de 2×2 de la imagen como sigue:

$I_{1,1}$	$I_{1,2}$
$I_{2,1}$	$I_{2,2}$

Entonces, una posible aproximación discreta al gradiente en dicha región es:

$$\frac{\partial f}{\partial x} = I_{1,2} - I_{1,1}$$

$$\frac{\partial f}{\partial y} = I_{2,1} - I_{1,1}$$

Otra posible alternativa para construir el operador de derivada en una máscara de 2x2 es tomar las diferencias cruzadas:

$$\frac{\partial f}{\partial x} = I_{1,1} - I_{2,2}$$

$$\frac{\partial f}{\partial y} = I_{1,2} - I_{2,1}$$

Donde $(\frac{\partial f}{\partial x})$ es el gradiente horizontal y $(\frac{\partial f}{\partial y})$ es el gradiente vertical. También podemos extender esta aproximación a un área de la imagen de 3×3, como sigue:

$I_{1,1}$	$I_{1,2}$	$I_{1,3}$
$I_{2,1}$	$I_{2,2}$	$I_{2,3}$
$I_{3,1}$	$I_{3,2}$	$I_{3,3}$

Aproximando el gradiente en este caso como:

$$\frac{\partial f}{\partial x} = (I_{3,1} + I_{3,2} + I_{3,3}) - (I_{1,1} + I_{1,2} + I_{1,3})$$

$$\frac{\partial f}{\partial y} = (I_{1,3} + I_{2,3} + I_{3,3}) - (I_{1,1} + I_{2,1} + I_{3,1})$$

Estas operaciones pueden ser implementadas mediante máscaras u operadores. En particluar, los últimos dos se conocen como los operadores de *Roberts* y *Prewitt*, y se implementan con máscaras de 2×2 y 3×3, respectivamente. Los máscaras se ilustran en las figuras 3.5 y 3.6.

1	0
0	-1

0	1
-1	0

Figura 3.5: Operadores de Roberts.

En la figura 3.7 se muestra el resultado de aplicar los operadores de Roberts y Prewitt. Las magnitudes se normalizaron entre 0 y 255 para mejorar el despliegue.

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Figura 3.6: Operadores de Prewitt.

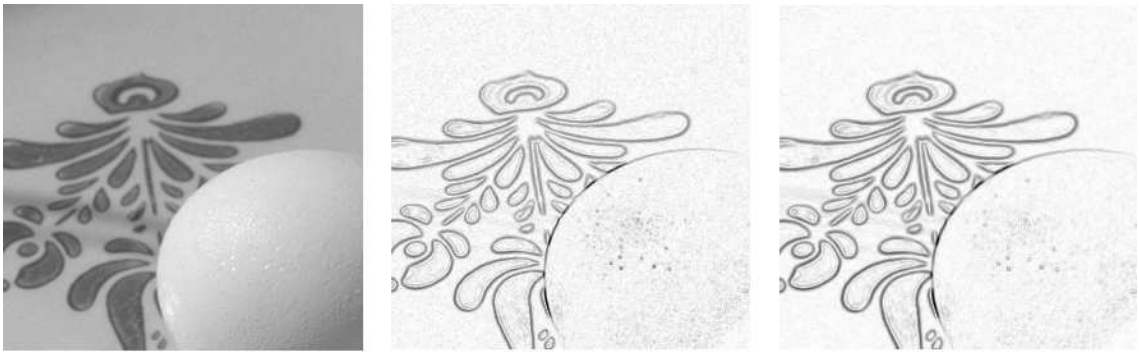


Figura 3.7: Detección de orillas con los operadores de Roberts y Prewitt. (a) Imagen original. (b) Magnitud resultante al aplicar los operadores de Roberts. (c) Magnitud resultante al aplicar los operadores de Prewitt.

3.2.1 Operadores de Sobel

Como se puede ver en la figura 3.7 los operadores de Roberts y Prewitt son sensibles al ruido. Para mejorar la detección de orillas podría utilizarse un preprocesamiento pare eliminar altas frecuencias o ruido. El detector de orillas *Sobel* incluye detección de orillas y suavizamiento. Los operadores de Sobel parten de los operadores de Prewitt adicionando ciertos pesos en la máscara que aproximan a un suavizamiento Gaussiano.

Los operadores de Sobel se pueden ver como la combinación unidimensioanl de diferenciación y cierto suavizamiento. Por ejemplo, uno de los operadores de Sobel lo podemos obtener como el producto de un vector de diferenciación (D) por uno de suavizamiento (G):

$$Sobel = \mathbf{DG^T} \tag{3.3}$$

Donde $D = (-1, 0, 1)$ y $G = (1, 2, 1)$. Esto reduce el efecto de amplificación del ruido que es característico de los operadores derivativos, por esto generalmente se prefiere el operador de Sobel a los anteriores. Los operadores de Sobel se pueden implementar con las máscaras que se ilustran en la figura 3.8. Un ejemplo de la aplicación de los operadores de Sobel a una imagen se ilustra en la figura 3.9.

3.2.2 Laplaciano de una Gaussiana

A finales de los 70s, David Marr estudio la visión de los mamíferos e ideó una teoría que integraba prácticamente todo lo que se conocía sobre la visión biológica. Su detector de orillas se basa en la segundas derivadas o Laplaciano de una Gaussiana. El Laplaciano de un función de dos variables se define como:

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figura 3.8: Operadores de Sobel. Observese el suavizamiento incluído a los operadores de Prewitt.

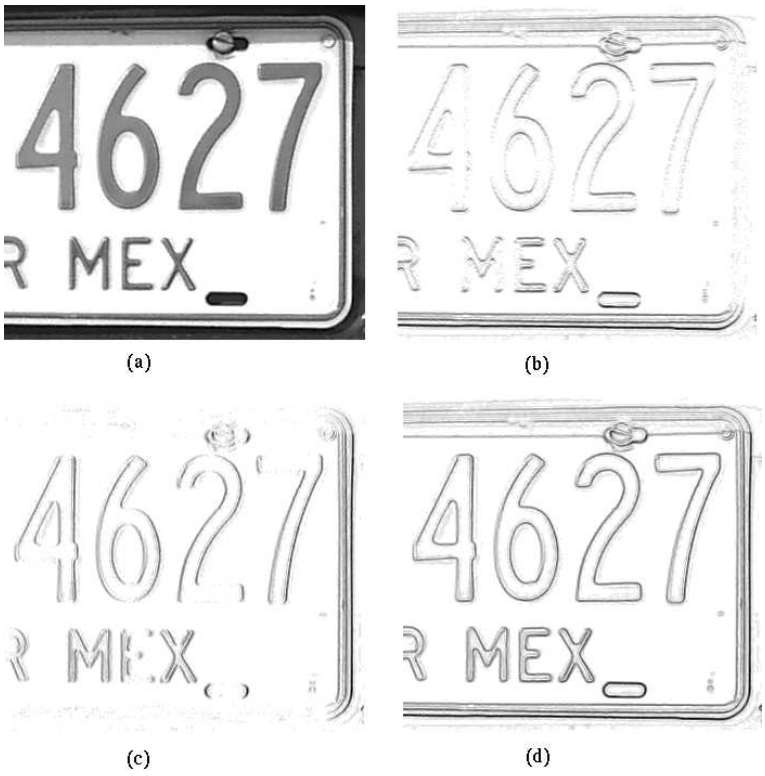


Figura 3.9: Detección de orillas con los operadores de Sobel. (a) Imagen original. (b) Valor absoluto del gradiente horizontal. (c) Valor absoluto del gradiente vertical. (d) Magnitud del gradiente. (Las magnitudes se normalizaron para mejorar el despliegue.)

$$\nabla^2 \mathbf{f} = \left(\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2} \right) \tag{3.4}$$

El cual se puede aproximar en forma discreta como:

$$\nabla^2 \mathbf{f} \approx 4 * I_{2,2} - I_{1,2} - I_{2,1} - I_{2,3} - I_{3,2} \tag{3.5}$$

La máscara correspondiente se muestra en la figura 3.10.

En una primera aproximación al Laplaciano de una Gaussiana, podría preprocesarse la imagen con un suavizamiento Gaussiano, para eliminar ruido, seguido de un operador Laplaciano. El Laplaciano de una Gaussiana (LOG: Laplacian of a Gaussian) se expresa como:

$$\nabla^2 \mathbf{G} = (\partial^2 G / \partial x^2) + (\partial^2 G / \partial y^2) \tag{3.6}$$

0	−1	0
−1	4	−1
0	−1	0

Figura 3.10: Máscara 3x3 para el operador Laplaciano.

Donde G es una distribución normal o Gaussiana en dos dimensiones.

La ventaja de usar un operador que se basa en la segunda derivada es que se puede estimar con mayor presición la localización de la orilla, que es *exactamente* donde la segunda derivada cruza cero. En la figura 3.11 se ilustra este efecto en una dimensión, donde se observa una función con un cambio repentino (orilla), la primera derivada y la segunada derivada donde se observa el cruce por cero. Nótese que para cada cambio repentino de la función, se genera un impulso que tiene cierto *ancho*, por lo que al aplicarse en imágenes se generan orillas dobles. Por lo anterior es necesario utilizar un postprocesamiento en donde se eliminen las dobles orillas.

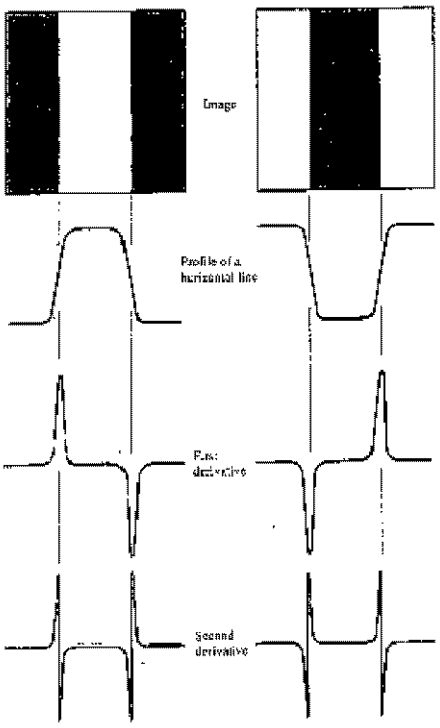


Figura 3.11: Cruce por cero de la primera y segunda derivada. De arriba a abajo: (a) imágenes, (b) perfil de una línea horizontal, (c) perfil de la primera derivada, (d) perfil de la segunda derivada.

En forma similar al operador Sobel, se puede combinar el efecto de un suavizamiento Gaussiano con el Laplaciano en una sola máscara. Una posible implementación se ilustra en la figura 3.12. La figura 3.13 muestra el resultado de aplicar este operador con una máscara de 3x3. La cantidad de falsas orillas que genera es considerable.

1	−2	1
−2	4	−2
1	−2	1

Figura 3.12: Operador "LOG": Laplaciano de una Gaussiana.

Otra manera de implementar un detector *LOG*, es diferenciar directamente dos Gaussianas,

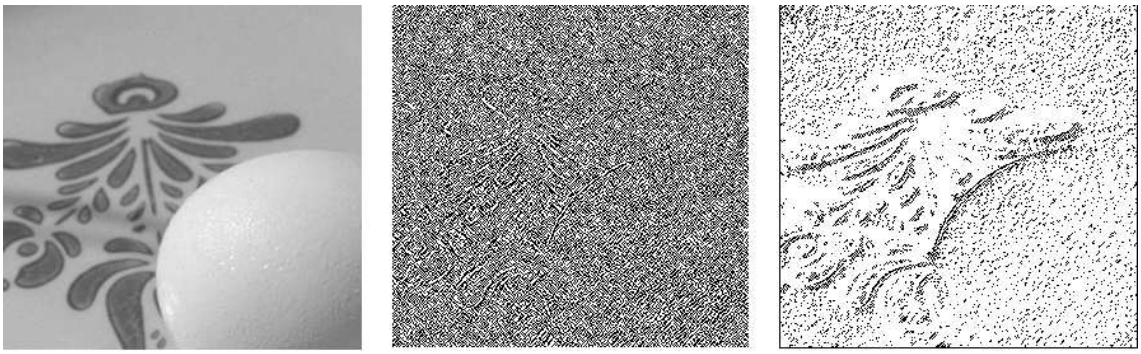


Figura 3.13: LOG utilizando la máscara de la figura 3.12. (a) Imagen original. (c) LOG utilizando máscara de 3x3. (d) Supresión de orillas dobles.

es decir, suavizar la imagen original en dos ocasiones (con distintas desviaciones estándar) para después restarlas. La figura 3.14 muestra la resta de dos Gaussianas[2].

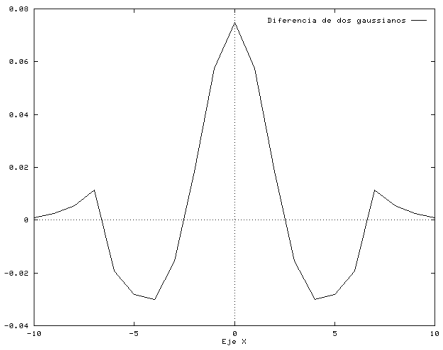


Figura 3.14: Aproximación al *LOG*: diferencia de dos Gaussianas.

En la figura 3.15 se muestra la salida de un detector de orillas tipo *LOG*. Obsérvese que en la imagen (*c*) y (*d*) se removieron las dobles orillas que ocasiona la segunda derivada. Una manera eliminar estas "falsas" orillas y orillas dobles es suprimir los puntos donde el gradiente no sea máximo en la dirección del borde, esto adelgaza la orilla ya que sólo permite tener un punto de alto gradiente a lo largo del borde. Esta técnica es conocida como supresión de no máximos (*non-maximum supression*). Detectores como *Canny* y *SUSAN* utilizan esta idea como postprocesamiento (ver sección de referencias).

Un problema de este operador es que no es posible obtener información de la direccionalidad de las orillas. En la siguiente sección veremos otros operadores que si manejan dirección.

Resultado de aplicar diversos operadores de detección de

3.3 Operadores direccionales

En general es necesario conocer no sólo la magnitud de las orillas sino también su direccionalidad. Esto es importante para los niveles superiores de visión, donde se desea unir las orillas en contornos y bordes. Para el caso del gradiente su dirección se define como:

[2]Marr recomienda utilizar una proporción de 1:1.6 entre las desviaciones estándar para obtener una buena aproximación al *LOG*.

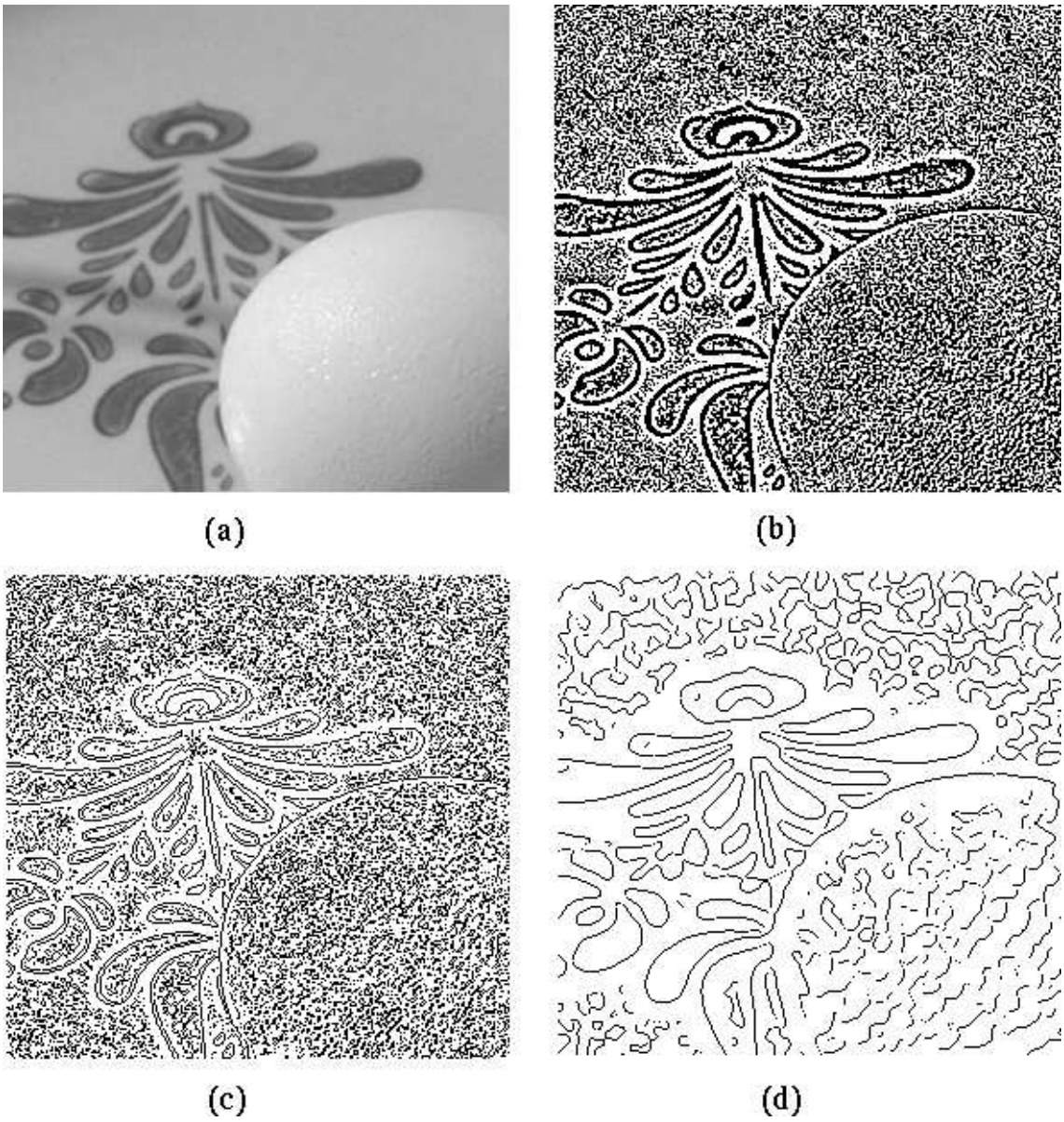


Figura 3.15: Laplaciano de una Gaussiana. (a) Imagen original. (b) DOG utilizando $\sigma_1 = 0.5$ y $\sigma_2 = 0.8$. Notese la presencia de orillas dobles. (c) Supresión de orillas dobles de la imagen anterior. (d) DOG utilizando $\sigma_1 = 2.5$ y $\sigma_2 = 4.0$ con supresión de orillas dobles.

$$\phi\mathbf{f} = tan^{-1}\left(\frac{(\frac{\partial f}{\partial y})}{(\frac{\partial f}{\partial x})}\right) \tag{3.7}$$

Entonces, podemos estimar la dirección de la orilla tomando la tangente inversa de los cocientes de los gradientes en x y y para los operadores de Prewitt y Sobel.

3.3.1 Operadores de Kirsch

Una generalización de los operadores de gradiente direccionales son las máscaras o *templates* de Kirsch. Los operadores de Prewitt detectan cambios en forma horizontal (0^o) y en vertical (90^o). Existen operadores que detectan orillas a más de dos diferentes orientaciones, como los *operadores*

de Kirsch. Los operadores de Kirsch son cuatro, de 0 a 135 grados, con 45 grados entre ellos, cuyo objetivo es detectar la dirección en que se tenga máxima respuesta, dando esto la direccionalidad de la orilla. Dichos operadores se pueden definir a diferentes tamaños, como $2x2$, $3x3$, $5x5$. Por ejemplo, las máscaras de los templates de Kirsch de $3x3$ se presentan en la figura 3.16.

−1	−1	−1
0	0	0
1	1	1

−1	−1	0
−1	0	1
0	1	1

−1	0	1
−1	0	1
−1	0	1

0	1	1
−1	0	1
−1	−1	0

Figura 3.16: Operadores de Kirsch en máscara de 3x3: 0, 45, 90 y 135 grados.

Dado que la respuesta tiene cierta dependencia en la magnitud de la función, y no sólo su derivada, es común utilizar máscaras de mayor tamaño ($5x5$) para reducir este efecto.

Dada la respuesta a cada operador a diferente dirección, se toma la orilla de mayor magnitud como la dirección de la orilla en cada pixel. La figura 3.17 muestra la magnitud de las orillas de una imagen para cada uno de los operadores de Kirsch ($3x3$).

3.3.2 Máscaras ortogonales de Frei-Chen

Como mencionamos anteriormente, un problema es saber si una orilla realmente es parte de un contorno (línea) o simplemente un punto aislado producto de otro fenómeno. Una forma de aproximarse a este objetivo fué propuesta por Frei y Chen y se basa en aplicar múltiples operadores simultáneamente a cada pixel y combinar los resultados.

Para comprender esta técnica, es conveniente considerar a los operadores como vectores, considerando su aplicación como un producto vectorial:

$$R = \sum_i w_i z_i \tag{3.8}$$

$$R = W^T Z \tag{3.9}$$

Donde W es el vector de pesos del operador, Z es el vector correspondiente a la imagen y R es el resultado de la aplicación del operador.

Si consideramos filtros de 2 elementos (bidimensionales), podemos pensar en dos vectores ortogonales y el vector de la imagen entre ellos. Entonces, el producto nos da la proyección del vector Z en cada uno de ellos. Si un filtro esta orientado a detectar orillas (diferencia de nivel entre dos regiones) y otro a detectar *líneas* (de un pixel de ancho), entonces la proyección relativa nos indica si el pixel se acerca más a uno u otro. Esto lo podemos ver gráficamente en la figura 3.18.

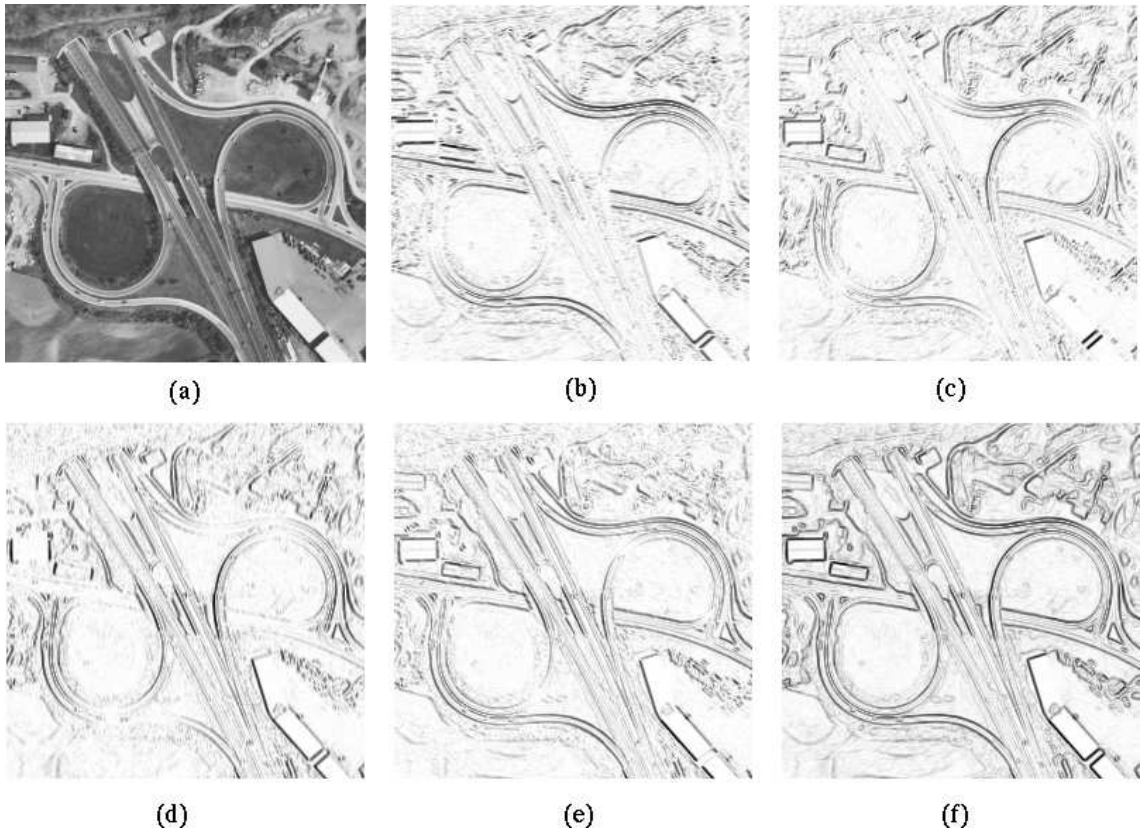


Figura 3.17: Resultado de aplicar los 4 operadores de Kirsch de 3×3 a una imagen. (a) Imagen original, (b) – (e) Magnitud de las orillas detectadas con los 4 operadores: 0, 45, 90 y 135 grados. (f) Se muestra el gradiente con mayor respuesta de las cuatro orientaciones.

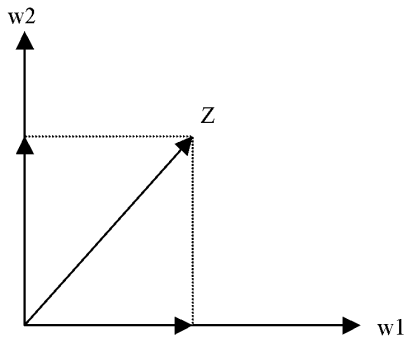


Figura 3.18: Proyección del vector. Si el vector Z representa a la imagen (en una región), y los vectores $w1$ y $w2$ a dos filtros (máscaras), la proyección de Z en cada uno corresponde a la magnitud resultante de aplicar el filtro correspondiente.

Este concepto lo podemos extender a otras bases y dimensiones, utilizando más tipos de detectores y de mayores dimensiones (tamaño). Un ejemplo de este tipo de operadores son las máscaras ortogonales de Frei-Chen, que se muestran en la figura 3.19. En este caso, 4 máscaras están enfocadas a detectar orillas, 4 a detectar líneas y una a detectar regiones de intensidad uniforme.

Para mejorar la información obtenida con las máscaras de detección de orillas, una alternativa es tomar la información de las orillas vecinas mediante una técnica iterativa denominada *relajación*.

Orillas

1	$\sqrt{2}$	1
0	0	0
-1	$\sqrt{2}$	-1

1	0	-1
$\sqrt{2}$	0	$-\sqrt{2}$
1	0	-1

0	-1	$\sqrt{2}$
1	0	-1
$-\sqrt{2}$	1	0

$\sqrt{2}$	-1	0
-1	0	1
0	1	$-\sqrt{2}$

Líneas

0	1	0
-1	0	-1
0	1	0

-1	0	1
0	0	0
1	0	-1

1	-2	1
-2	4	-2
1	-2	1

-2	1	-2
1	4	1
-2	1	-2

Uniforme

1	1	1
1	1	1
1	1	1

Figura 3.19: Máscaras ortogonales de Frei-Chen.

3.4 Relajación

Una forma de mejorar los detectores de orillas es tomar en cuenta la información de los pixels vecinos (figura 3.20). Si consideramos que la orilla constituye parte de un borde o contorno mayor, entonces existe una alta probabilidad que las orillas se encuentren contíguas; en cambio, si es un elemento aislado producto del ruido u otra causa, entonces es poco probable que existan otras orillas a su alrededor.

Una técnica iterativa que hace uso de este tipo de información se conoce como relajación. Relajación consiste, escencialmente, de una serie de etapas de la siguiente forma:

1. Obtener una estimación inicial de las orillas y su *confidencia*.

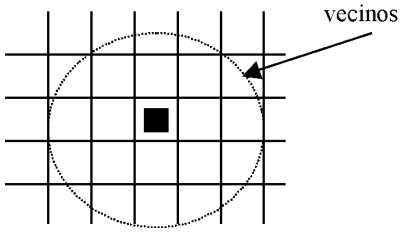


Figura 3.20: Un esquema de vecindad.

2. Actualiza la magnitud de la orilla en base a sus vecinos.
3. Actualiza la confidencia de la orilla.
4. Repite 2 y 3 hasta que se cumpla cierto criterio de terminación o hasta llegar al máximo de iteraciones.

Existen varios algoritmos para calcular la confidencia y actualizar las orillas. Una alternativa es el método propuesto por Prager. El algoritmo de Prager se basa en una clasificación de tipos de orillas y a partir de éstos, se definen fórmulas para calcular y actualizar su confidencia. Las orillas se clasifican a partir del número de orillas que existen en los vecinos de un *vértice* de la orilla de interés. Los vértices de una orillas son los extremos, izquierdo-derecho o superior-inferior de la orillas. Existen varios tipos de vértices los cuales se ilustran en la figura 3.21.

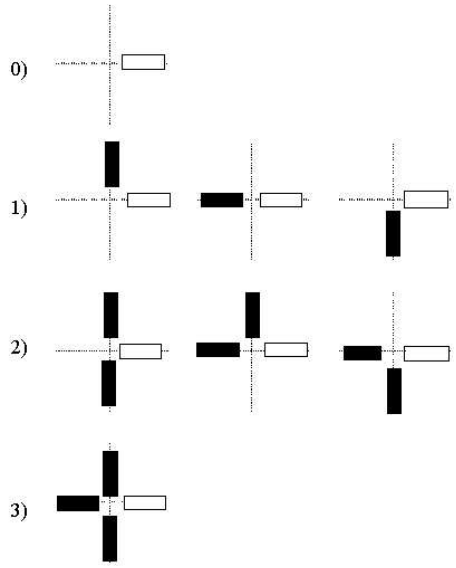


Figura 3.21: Tipos de vértices: 0) vértice con 0 orillas vecinas, 1) vértices con 1 orilla vecina, 2) vértices con 2 orillas vecinas, 3) vértice con 3 orillas vecinas.

De acuerdo a esto existen 4 tipos de vértices y su confidencia se calcula de acuerdo al tipo, de la siguiente forma:

- (ninguna orilla) $C(0) = (m-a)(m-b)(m-c)$
- (1 orilla) $C(1) = a(m-b)(m-c)$
- (2 orillas) $C(2) = ab(m-c)$
- (3 orillas) $C(3) = abc$

Donde:

- a, b, c son las magnitudes (normalizadas) de las orillas vecinas,
- $m = max(a, b, c, q)$,
- q es una constante entre 0 y 1.

Se considera el tipo de vértice "j" de forma que $C(j)$ sea máxima.

El tipo de orilla (ij) es la concatenación del tipo de sus dos vértices. Para actualizar la confidencia se basa en el tipo de orilla y se usan las siguientes ecuaciones:

- Tipos (11,12,13), incrementar: $C(k+1) = min(1, C(k) + d)$
- Tipos(00,02,03), decrementar: $C(k+1) = max(0, C(k) - d)$
- Tipos(01,22,23,33), dejar igual: $C(k+1) = C(k)$

Donde d es una constante que controla la rapidéz de convergencia del método (normalmente entre 0.1 y 0.3).

Para aplicar este método se utiliza algún detector de orillas (Sobel, Prewitt, etc.) para obtener una estimación incial, utilizando la magnitud de la orilla como un estimado de la confidencia inicial. El proceso de repite un número determinado de veces o hasta que el número de cambios en una iteración sea menor a un umbral predefinido. Generalmente se obtienen buenos resultados. El principal inconveniente es que el proceso es costoso computacionalmente (iterativo).

3.5 Comparación de operadores

Como se menciono al inicio del capítulo, detección de orillas es un tema que ha generado una gran cantidad de publicaciones científicas. Esto es debido, a diversas maneras de como definir lo que es *una orilla*. Se sabe que el problema es complejo, ya que para una misma imagen se pueden generar más de una imagen de orillas como resultado válido (no existe una solución única)[3]. Basados en esta subjetividad se han propuesto una gran cantidad de algoritmos, donde cada uno indica que es "óptimo" en algún sentido.

La comparación y selección entre detectores de orillas se ha convertido en una tarea compleja. Una métrica o "figura de mérito" para tratar de compararlos "objetivamente" es la siguiente:

$$F = \frac{1}{max(NA, NI)} \sum_i \frac{1}{1 + ad_i^2} \tag{3.10}$$

Donde:

- NA - num. de orillas detectadas
- NI - num. de orillas "ideales" o reales.
- d - distancia entre las orillas detectadas e ideales
- a - constante

[3]Es lo que se conoce como un problema mal planteado [120], en el sentido de Hadamard [30].

Experimentalmente se ha incontrado que todos los operadores tiene medidas similares, y que su respuesta se va deteriorando de acuerdo a la cantidad de ruido en la imagen. En la figura 3.22 se grafica en forma aproximada la respuesta de diferentes operadores en función de la cantidad de ruido (razón de señal a ruido) de la imagen.

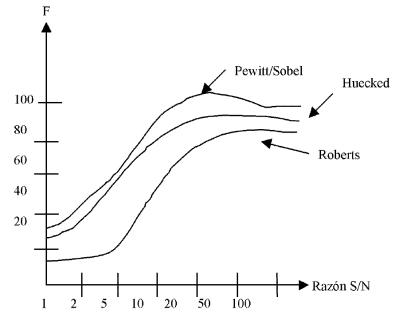


Figura 3.22: Comparación de diferentes operadores.

De acuerdo ha esto no tiene, en general, mucha importancia el tipo de operador seleccionado para detección de orillas, y los problemas deben resolverse en los niveles superiores de visión.

En la práctica, generalmente se establece un límite inferior (*threshold*) para considerar las orillas detectadas, eliminando todas las que sean menores a este límite (éste puede ser entre 10 y 30% de la magnitud máxima).

3.6 Referencias

La detección de orillas es una de los aspectos que más se ha investigado en visión. Entre los trabajo iniciales en detección de orillas se encuentran los de Roberts [98] y Prewitt [93]. También destacan los desarrollos de Marr [76, 77], quien estudia los fundamentos teóricos de la detección de orillas y su implementación biológica. Los otros detectores que se describen en el capítulo se basan en los trabajo de [61], [24] y [92]. Hay detectores de orillas más sofisticados, como el de Canny [10] y el *Susan* [109].

Entre las nuevas técnicas podremos comentar los algoritmos de "*edge sharpening*", del tipo difusión anisotrópica [89, 90] y derivados [14, 99], los cuales facilitan la detección de orillas. Básicamente estos algoritmos realizan, de manera iterativa, un suavizamiento donde las orillas "más significativas" se preservan y el resto de la imagen se suaviza. La decisión sobre suavizar o no se toma en base a una función que implícitamente contiene un operador de derivada. Después de la etapa de suavizamiento, la detección de orillas se convierte en una tarea fácil ya que las principales discontinuidades tendrán un gradiente "significativamente" mayor que el fondo. Una posterior umbralización completa la detección de orillas. Los problemas asociados a esta familia de técnicas es la determinación del número "suficiente" de iteraciones o suavizamiento, ya que en la convergencia (un número grande de iteraciones) la imagen se convierte en homogénea perdiendo todos los atributos útiles.

Otros autores han utilizado el espacio de escalas para localizar las orillas más significativas. En [5, 74] las definen como las que "sobreviven" a cierto nivel de suavizamiento (generalmente en $\sigma = 5$ o mayor). Esta definición de orillas significativas ha demostrado ser incorrecto por varias razones. Una orilla que se mantiene a una escala tan grande es provocado por un gran contraste en la imagen original y no necesariamente se refiere a una orilla "significativa". Además, por el mismo suavizamiento, las orillas se mueven y unen a través del espacio de escalas; es decir, la orilla que veamos a una escala grande puede *no existir* en la imagen original. Para encontrar la orilla original es necesario hacer un seguimiento de la orilla hacia atras en las escalas, lo cual ha resultado ser un problema mal planteado (ill-posed).

Lindeberg ha publicado una definición de orilla que, de alguna manera, incluye al *non-maximum suppression*. Lindeberg define que una orilla es el lugar donde se genera un cruce por cero de la segunda derivada y el signo de la tercera derivada es negativo. Escrito de manera matemática: $I_{vv} = 0$ y $I_{vvv} < 0$. Este detector de orillas es bastante sensible al ruido (por tener una segunda derivada) y necesita de un postprocesamiento más complicado.

El integrar orillas en bordes, que será tratado detenidamente en el capítulo de visión de nivel intermedio, normalmente se realiza como una doble umbralización o *hysteresis*, en donde las orillas mayores a cierto umbral t_{max} se marcan instantáneamente como orillas, mientras que las orillas mayores a t_{min} se analizan verificando que formen un borde y eliminando las orillas aisladas. Los detectores de orillas Canny [10], Lindeberg [69] y Shen-Castan [105] utilizan esta técnica.

En la práctica, la detección de orillas se realiza en más de una etapa. Normalmente se encuentran combinaciones de *non-maximum suppression*, verificación de signos en la tercera derivada, superposición de primera y segunda derivada, *hysteresis*, suavizamiento, etc. Otra alternativa es utilizar técnicas de relajación, como el método propuesto por Prager [92]; o métodos de regularización basados en Campos de Markov.

3.7 Problemas

1. En el diseño de las máscaras para detección de orillas, un aspecto a considerar es el tamaño de la máscara. ¿Qué impacto tiene esto en la capacidad de detección de orillas? ¿Qué compromisos hay respecto al tamaño de la máscara?

2. ¿Qué diferencia hay entre los operadores de gradiente (Prewitt, Sobel) y el laplaciano para detección de orillas? ¿Qué ventajas tienen los dos diferentes enfoques?

3. Demuestra que el valor promedio de cualquier imagen a la que se le aplique (por convolución) el operador laplaciano es cero.

4. ¿Qué diferencia hay entre los operadores de primera derivada y de segunda derivada para la detección de orillas? ¿Qué ventajas tiene cada uno de los dos enfoques? Da un ejemplo de un operador de c/u.

5. Considera la siguiente imagen (binaria):

0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

   Da el resultado (como imágenes) de aplicar los operadores de Sobel, incluyendo cada uno por separado y la magnitud combinada. Especifica que consideraste para el "borde" de la imagen.

6. Dada la siguiente imagen, obten la magnitud de las orillas aplicando el operador laplaciano y muestra la imagen resultante. Especifica que consideraste para el "borde" de la imagen.

1	1	1	0
0	1	1	1
0	0	1	1
0	0	0	1

7. Considera 3 tipos de orillas:

   (a) escalón,

   (b) rampa,

(c) cresta (línea).

Obten la respuesta de los operadores de Prewitt, Sobel y Laplaciano para c/u. Comenta los resultados obtenidos.

8. Especifica la dirección de las líneas para las que obtendrían una respuesta mayor los 4 operadores de "Líneas" de las máscaras ortogonales de Frei-Chen.

9. Propon un algoritmo de relajación que tome como partida el operador de Sobel y que base su confidencia y actualización en información de la dirección de la orilla.

10. Escribe en "pseudo–código" una rutina para obtener las orillas en una imagen utilizando relajación. Considera que ya se tiene la imagen en una arreglo E de $n \times n$ y el que resultado se almacena en un arreglo S de la misma dimensión. Describe las variables y constantes que utilizes en el programa.

3.8 Proyectos

1. Implementar en en el laboratorio un detector de orillas utilizando el operador laplaciano. Desplegar la salida, probando con diferentes imágenes.

2. Implementar en en el laboratorio un detector de orillas utilizando las máscaras de Sobel (en X, Y), obtener magnitud en dos formas diferentes: absoluto y máximo. Desplegar la salida en X, en Y y la magnitud. Probar con diferentes imágenes.

3. Para los detectores de orillas de los proyectos anteriores, probar que diferencias hay en la salida si la imagen se filtra (pasa-bajos) o ecualiza previamente.


(a) (b) (c)

Capítulo 4

Procesamiento del color

4.1 Introducción

El utilizar color en visión es importante ya que puede ayudar a la extracción de características e identificación de objetos en la imagen, lo cual, en ciertos casos, puede ser muy difícil en imágenes monocromáticas. El ojo humano puede distinguir miles de colores (con diferentes intensidades y saturaciones) y en cambio sólo distingue alrededor de 20 niveles de gris. Por esto se piensa que el color tiene un papel muy importante en el reconocimiento.

La percepción del color en el ser humano es un proceso psicofisiológico que aún no es bien comprendido. El color que percibe el ser humano de un objeto depende de la naturaleza de la luz reflejada por el objeto, lo que a su vez depende de la luz incidente en el objeto.

Físicamente, la luz visible es parte del espectro electromagnético, y el color tiene que ver con la longitud de onda dentro del espectro visible (400 - 700 nm). La luz blanca consiste de la combinación de todos los colores en dicho espectro, el cual se muestra en la figura 4.1.

color:	violeta	azul	verde	amarillo	naranja	rojo
longitud de onda (nm):	400					700

Figura 4.1: Espectro electromagnético del rango visible y los principales rangos de colores asociados.

Un objeto se ve de cierto color bajo una luz "blanca", si refleja la luz de longitudes de onda alrededor de dicho color (ej. verde = 500-570) y absorbe el resto de las longitudes de onda. El observador (o una cámara) percibe el color del objeto en función de las longitudes de onda que el objeto refleja (figura 4.2).

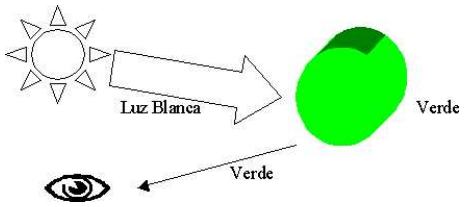


Figura 4.2: Percepción del color. Al ser iluminado un objeto con luz blanca, éste absorbe ciertas longitudes de onda y refleja otras. El color que percibimos depende de la longitud de onda dominante de la energía reflejada.

Dicho objeto puede no tener un color "puro" (saturado), sino que también refleje luz a otras longitudes de onda, tal vez con menor intensidad. Dicha luz reflejada puede tener diferente intensidad o brillantez dependiendo de la luz incidente y la naturaleza del objeto.

En base a lo anterior podemos distinguir tres atributos básicos del color:

- longitud de onda dominante o croma (*Hue*),
- pureza o saturación,
- brillantez o intensidad.

4.2 Percepción de color

El ser humano percibe el color mediante unos sensores (conos) que traducen la energía lumínica incidente en señales nerviosas que van a la parte visual del cerebro. Estos están concentrados en la parte central de la retina y se pueden dividir en 3 clases, dependiendo de la banda de longitudes de onda a la cual son más sensibles. Los sensores tipo α tienen una mayor sensibilidad a 480 nm (azul), los tipo β a 540 nm (verde) y los tipo γ a 570 nm (rojo). Esta información se resume en la figura 4.3. Nótese que la banda de sensitividad de dichos receptores se traslapa.

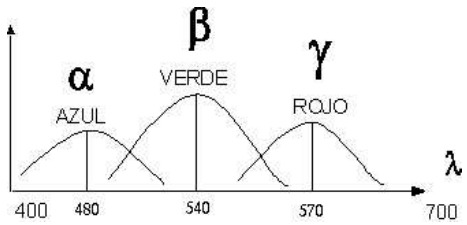


Figura 4.3: Respuesta del ojo humano a diferentes longitudes de onda.

La identificación de la información cromática (color) de la imagen se hace mediante la combinación de estas 3 señales, de dónde se perciben la gran variedad de colores que podemos distinguir. A estos se les denomina colores primarios (rojo, verde y azul). De la combinación aditiva en partes iguales de éstos, en pares, obtenemos los colores secundarios (amarillo, magenta, cian); y de los 3, el blanco. Otra forma es combinar los secundarios substractivamente de donde obtenemos los primarios y negro (figura 4.4).

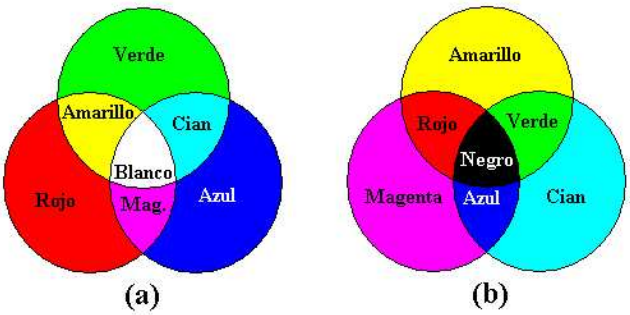


Figura 4.4: Diagrama cromático para el sistema RGB. (a) Mezclas de luz. Adición de primarios. (b) Mezcla de pigmentos. Substracción de secundarios.

Por ejemplo, la televisión se base en la combinación aditiva y las pinturas o el uso de filtros en la substractiva. En la figura 4.5 se ilustra una imagen a color y las imágenes de sus 3 componentes primarios.

4.3 Sistema CIE

La hipótesis de que todos los colores pueden ser generados de una combinación de los tres primarios ha sido comprobada experimentalmente mediante lo que se conoce como "apareamiento de colores" (*color matching*). Para ello se presenta a un observador dos campos contiguos con los siguientes colores:

- Una luz monocromática a cierta longitud de onda.
- Una luz que es combinación de tres luces primarias a ciertas longitudes de onda conocidas.

El observador ajusta la intensidad de los primarios hasta que las dos partes se "ven" iguales, es decir, que el "match" es psico-fisiológico. Entonces se tiene que un color se obtiene como una mezcla de diferentes proporciones de los 3 primarios:

$$C = k_1 R + k_2 G + k_3 B \tag{4.1}$$

Esto se realizó para toda la gama de colores visibles (cada 5 nm, por ejemplo), obteniéndose k_1, k_2 y k_3. Por ejemplo, la transformación de una imagen RGB a monocromática, M, se hace con los valores de k siguientes:

$$M = 0.33R + 0.5G + 0.17B \tag{4.2}$$

Un ejemplo de esta fórmula es la conversión de la imagen 4.5-a en su correspondiente imagen monocromática 4.5-e.

Una observación muy importante es que ciertos colores no se lograban igualar con ninguna combinación de los 3 primarios. Para lograr la igualación, se suma a algún primario al color a igualar, lo que equivale a una componente negativa de dicho primario. De esta forma se obtuvieron las funciones de igualación para el sistema RGB.

Si se normalizan los valores de R, G, B de forma que sumen uno, obtenemos lo que se conoce como coordenadas cromáticas:

$$r = R/(R + G + B) \tag{4.3}$$

$$g = G/(R + G + B) \tag{4.4}$$

$$b = B/(R + G + B) \tag{4.5}$$

Por lo tanto:

$$r + g + b = 1, b = 1 - r - g \tag{4.6}$$

De forma que el espacio de colores lo podemos representar en 2 dimensiones (r y g, por ejemplo) en un *diagrama cromático* (figura 4.6). Entonces el tercer color primario (b) queda implícito, ya que suma uno.

La "Comisión Internacional de Iluminación" (CIE) estandarizó como colores primarios: azul = 435.8 nm, verde = 546.1 nm, rojo = 700 nm, que corresponden a las primarias denominadas X, Y,

Figura 4.5: Componentes de una imágen a color. (a) Imagen original. (b) Componente en rojo. (c) Componente en verde. (d) Componente en azul. (e) Transformación a monocromática.

Z, y las correspondientes coordenadas cromáticas x, y, z. El objetivo de los primarios seleccionados es evitar las componentes negativas. Graficando en dos dimensiones, $x - y$, obtenemos la figura 4.7. Éste diagrama cromático tiene varias propiedades importantes:

- El perímetro representa todos los colores "puros" o completamente saturados.
- Los puntos interiores tienen cierta proporción de los 3 colores (blanco).
- El punto de la misma energía de los 3 primarios corresponde al blanco.
- La línea que une 2 punto nos da todas las combinaciones que se pueden formar a partir de 2 colores.
- El triángulo que forman los tres puntos nos da todos los colores que se pueden obtener de la combinación de los tres básicos.

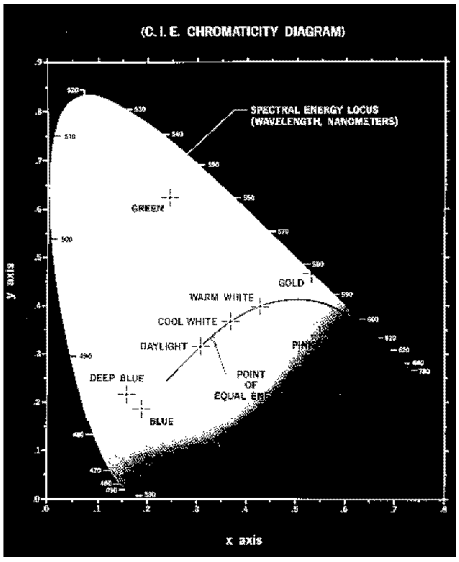


Figura 4.6: Diagrama cromático CIE: normalización del diagrama cromático en dos dimensiones.

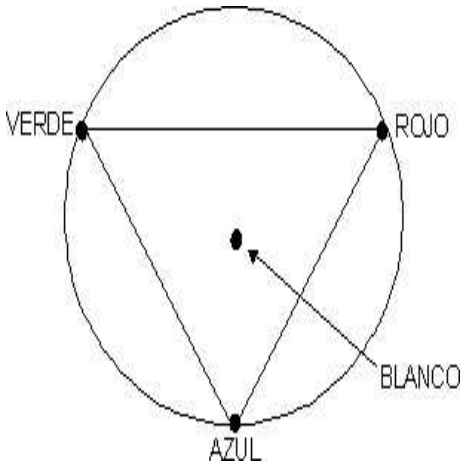


Figura 4.7: Diagrama en dos dimensiones del sistema RGB.

De esta última propiedad se ve que no es posible obtener todos los colores de la combinación de 3 primarios, ya que habrá partes del diagrama que queden fuera del triángulo.

Además de diagramas cromáticos como el de CIE, existen otras formas de representar el color que son más adecudas para diferentes aplicaciones, incluyendo visión y procesamiento de imágenes, llamados *modelos de color*.

4.4 Modelos de color

Existen varias representaciones o modelos de color. Estos modelos los podemos dividir en dos clases de modelos. Unos son los modelos que están más orientados a los equipos, por ejemplo las cámaras o monitores de televisión, a los que llamaremos *modelos sensoriales*. Otros son los modelos que se asemejan más a la percepción humana y que, en general, están orientados al procesamiento de imágenes y visión, éstos se denominan *modelos perceptuales*.

4.4.1 Modelos Sensoriales

Dentro de los modelos sensoriales de color existen 3 modelos más comunmente utilizados: RGB, CMY e YIQ.

Modelo RGB

El modelo RGB es el modelo básico que utiliza las componentes primarias rojo, verde y azul, normalizadas. De esta forma los colores se representan en coordenadas cartesianas dentro de un cubo unitario (figura 4.8).

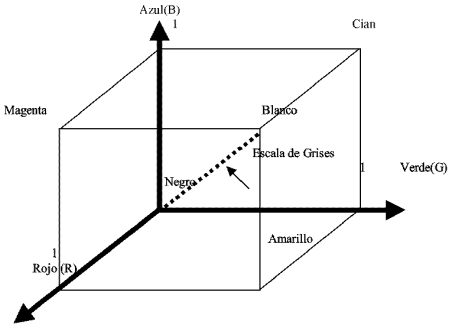


Figura 4.8: Cubo unitario de color para el modelo RGB.

Cada color se representa como un vector del origen y la diagonal principal corresponde a la escala de grises. En este modelo se basan las cámaras y receptores de televisión (TV). Sin embargo, se tienen problemas al aplicarlo a procesamiento de imágenes (ecualización) y visión (no-lineal), como veremos en las siguientes secciones.

Modelo CMY

El modelo CMY se basa en los colores secundarios (cian, magenta, amarillo). Este se puede obtener del modelo de RGB de la siguiente forma:

$$\begin{matrix} C \\ M \\ Y \end{matrix} \quad = \quad \begin{matrix} 1 \\ 1 \\ 1 \end{matrix} \quad - \quad \begin{matrix} R \\ G \\ B \end{matrix} \tag{4.7}$$

Se usa este modelo al combinar colores (depósito de segmentos) en papel, como en impresoras y copiadoras de color.

Modelo YIQ

En el modelo YIQ se separa la información de intensidad o luminancia (Y) de la información de color (I, Q). Se obtiene mediante la siguiente transformación a partir de las componentes del RGB:

$$\begin{matrix} Y \\ I \\ Q \end{matrix} \quad = \quad \begin{matrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.231 \\ 0.212 & -0.523 & 0.311 \end{matrix} \quad \begin{matrix} R \\ G \\ B \end{matrix} \tag{4.8}$$

Este es el sistema que se utiliza para la transmisión de TV a color. Tiene dos ventajas: (i) la separación de la información de luminancia para compatibilidad con receptores de blanco y negro y, (ii) el uso de mayor ancho de banda (bits) para esta información que es más importante para la percepción humana.

4.4.2 Modelos perceptuales

Los sistemas anteriores están más orientados a los equipos, mientras que los siguientes modelos, llamados modelos perceptuales, tienen cierta similitud con la percepción humana, por lo que están más enfocados a visión. Éstos sistemas, generalmente, utilizan una representación en base a los parámetros perceptuales: croma (*Hue*, H), saturación (S) e intensidad (I).

Modelo HSV

El modelo "HSV" (*Hue, Saturation, Value*) se obtiene "deformando" el cubo RGB de forma que se convierte en una pirámide hexagonal invertida. En el vértice se tiene el negro, en las esquinas del hexágono los 3 primarios y secundarios y en su centro el blanco. El modelos HSV se ilustra en forma geométrica en la figura 4.9.

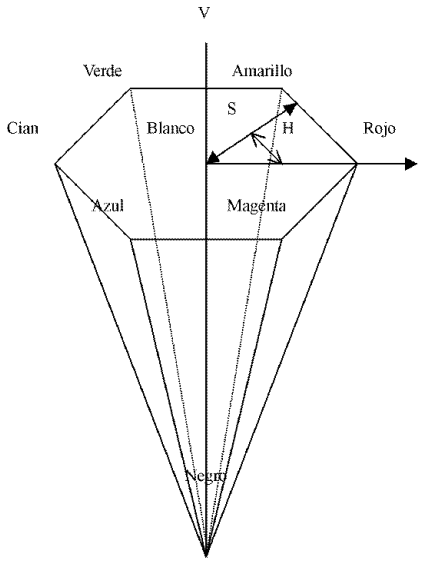


Figura 4.9: Modelo de color HSV.

De esta forma el eje vertical representa la brillantez o valor (V), el horizontal la saturación (S) y el ángulo de la proyección horizontal el croma (H). La conversión de RGB a HSV se logra mediante las siguientes ecuaciones:

$$V = M; [0, 1] \tag{4.9}$$

$$Si : M = m, S = 0; sino, S = (M - m)/M; [0, 1] \tag{4.10}$$

$$Si : m = B, H = 120(G - m)/(R + G - 2m); [0, 360] \tag{4.11}$$

$$Si : m = R, H = 120(B - m)/(B + G - 2m); [0, 360] \tag{4.12}$$

$$Si : m = G, H = 120(R - m)/(R + B - 2m); [0, 360] \tag{4.13}$$

Donde $m = Min(R, G, B)$ y $M = Max(R, G, B)$. La brillantez (V) y saturación (S) están normalizada (entre cero y uno) y el croma (H) esta entre 0 y 360 grados.

Modelo HLS

El modelo HLS (*Hue, Level, Saturation*) se basa en coordenadas polares en 3 dimensiones, obteniéndose un espacio en forma de 2 conos unidos en su base. El vértice inferior corresponde a negro, el superior a blanco; el eje vertical representa la brillantez (L), el horizontal la saturación (S) y el ángulo de la proyección horizontal el croma (H). El espacio geométrico del modelo HLS se muestra en la figura 4.10.

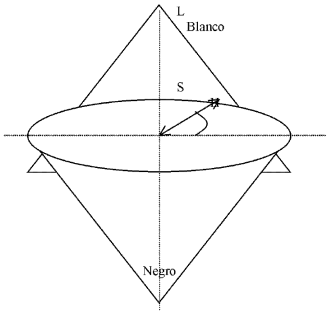


Figura 4.10: Modelo de color HLS.

La transformación del modelo RGB al HLS se obtiene con las siguientes ecuaciones:

$$L = (M + m)/2 \tag{4.14}$$

$$S = (M + m)/(M - m), siL \leq 0.5 \tag{4.15}$$

$$S = (M - m)/(2 - M - m), siL > 0.5 \tag{4.16}$$

H: igual al modelo HSV

Donde $m = Min(R, G, B)$ y $M = Max(R, G, B)$. La brillantez (L) y saturación (S) están normalizada (entre cero y uno) y el croma (H) esta entre 0 y 360 grados.

Modelo HSI

El modelo HSI (*Hue, Saturation, Intensity*) se puede ver como una transformación del espacio RGB al espacio perceptual. Tiene una forma de dos pirámides triangulares unidas en su base. Los vértices de las pirámides corresponden a blanco y negro, y los del trángulo a R, G, B (éste es análogo al triángulo del diagrama cromático). En forma similar a los modelos anteriores, la intensidad (I) se mide en el eje vertical, la saturación (S) en función a la distancia a este eje y el croma (H) como el ángulo horizantal tomado el rojo como referencia (cero grados). El modelo se ilustra en la figura 4.11.

La transformación de RGB a HSI se realiza mediante la siguientes ecuaciones:

$$H = cos^{-1}\left(\frac{\frac{1}{2}(R - G) + (R - B)}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right) \tag{4.17}$$

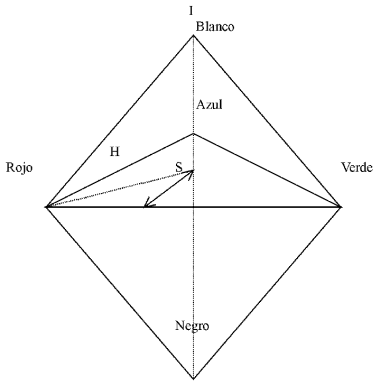


Figura 4.11: Modelo de color HSI.

$$S = 1 - \left(\frac{3min(R,G,B)}{R+G+B} \right) \tag{4.18}$$

$$I = \frac{1}{3}(R+G+B) \tag{4.19}$$

si $B > G : H = 2\pi - H$.

La intensidad (I) y saturación (S) están normalizada (entre cero y uno) y el croma (H) esta entre 0 y 360 grados. Un ejemplo de una imagen en el modelo de color HSI se ilustra en la figura 4.12.

Figura 4.12: Ejemplo de imagen en el modelo de color HSI. (a) Croma H. (b) Saturación S. (c) Intensidad I.

4.4.3 Comparación entre modelos

Desde el punto de vista de visión, los aspectos importantes a evaluar en los diferentes modelos son:

- Linearidad: que exista una relación lineal entre los atributos del color y la percepción del color.
- Uniformidad: que el espacio de color sea unifrome en cuanto a su correspondencia con la percepción del color.
- Singularidades: que no existan singularidades en el espacio, es decir, puntos donde haya cambios bruscos en cuanto a la relación con la percepción del color.
- Analogía a la percepción humana: que el modelo se asemeje a la forma en que los humanos percibimos el color.

4.5 Pseudo–color

La diversas técnicas de *pseudo–color* están orientadas al procesamiento de imágenes monocromáticas para facilitar su interpretación visual. En general consisten en hacer algun tipo de transformación de niveles de gris a colores.

4.5.1 Partición de intensidades

Si consideramos la imagen monocromática como una función tridimensional, podemos dividirla mediante planos en diferentes regiones, asignando un color diferente a cada "rebanada". Esta técnica se conoce como *partición de intensidades* y se puede ilustrar en forma gráfica como se muestra en la figura 4.13.

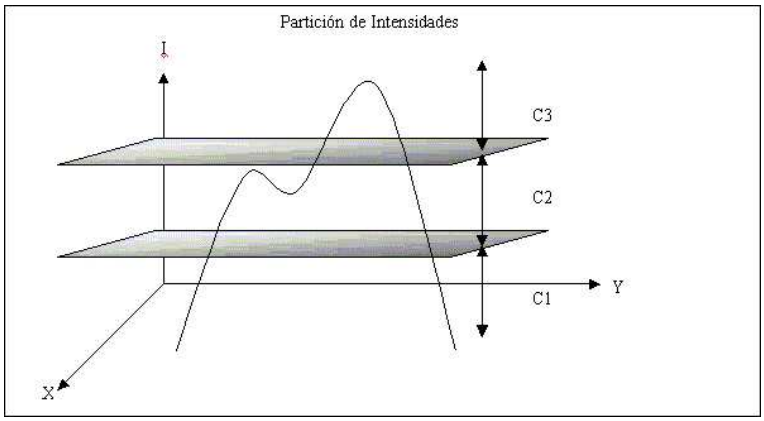


Figura 4.13: Partición de intensidades.

Para esto se divide el rango de niveles de gris en un número n de rangos, definiendo una serie de $n-1$ umbrales entre cada rango. Para cada rango se selecciona un color y todos los pixels en dicho rango se transforman al color correspondiente. En la figura 4.14 se muestra un ejemplo de una imagen en tonos de gris que ha sido transfromada a un imagen a color mediante este procedimiento.

4.5.2 Transformación de nivel de gris a color

Consiste en aplicar tres transformaciones diferentes a los niveles de gris, y cada una aplicarla a los colores primarios –R, G, B–, de forma que se obtiene una imagen a color combinándolos. Un diagrama de bloques de este proceso se ilustra en la figura 4.15.

Las funciones de transformación pueden ser, en principio, cualquier función lineal o no-lineal, que realize un mapeo del nivel de gris a cada uno de las componentes de color. La definición

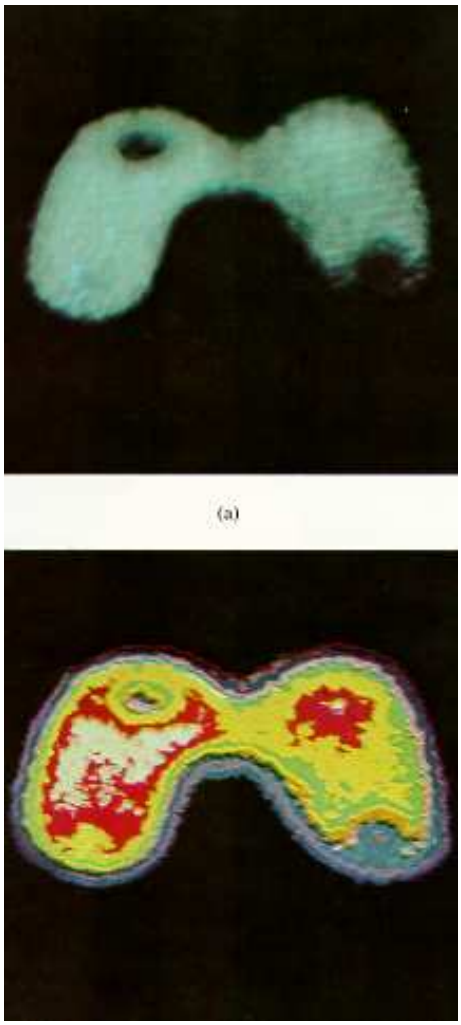


Figura 4.14: Transformación de una imagen mediante partición de intensidades: (a) imagen original monocromática, (b) imagen a color resultante utilizando ocho rangos.

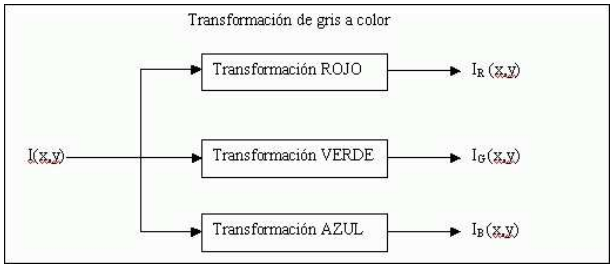


Figura 4.15: Transformación de gris a color.

de estas funciones dependería de el tipo de imagen, de forma que se obtengan diferentes colores para los diferentes objetos que se desean distinguir. La figura 4.16 muestra una posible función de transformacion del nivel de gris a la componente R (rojo). La técnica de partición de intensidades entonces puede considerarse como un caso especial de una transformación de gris a color, en el cual las funciones de transformación son lineales a pedazos (como una función tipo *escalera*).

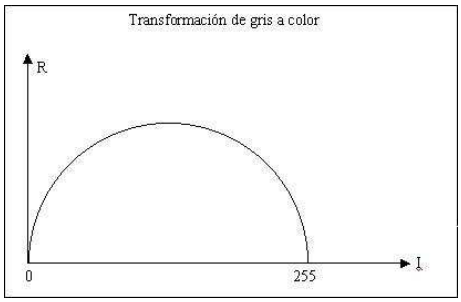


Figura 4.16: Ejemplo de una función de transformación de gris a color.

4.5.3 Transformación en frecuencia

La transformación a color en frecuencia es análoga a la transformación de gris a color, pero en este caso se toma la transformada de Fourier de la imagen monocromática y se le aplican las transformaciones en el dominio de la frecuencia. De esta forma se pueden aplicar filtros diferentes para R, G, B; haciendo un mapeo de frecuencias espaciales a color. El proceso se ilustra en forma de diagrama de bloques en la figura 4.17. Para ello se obtiene la transformada de Fourier de la imagen, luego se aplican diferentes filtros para cada componente de color y, finalmente, se obtiene la transformada inversa de cada uno para integrar la imagen en pseudo–color resultante.

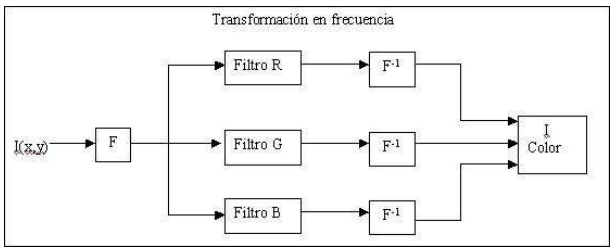


Figura 4.17: Transformación en frecuencia.

Aplicando esta técnica se pueden observar áreas de diferente frecuencia de la imagen original a diferentes colores. Por ejemplo, se le podrían asignar las bajas frecuencias (regiones uniformes) a un color y las altas frecuencias (regiones de cambio u orillas) a otro color.

4.6 Procesamiento de Imágenes a Color

Trabajando ahora directamente en la imagen a color, el objetivo es mejorarla para su interpretación visual o para alimentarla a los niveles superiores de visión. Normalmente se pueden aplicar las técnicas para imágenes monocromáticas a imagenes a color, aplicándose a cada componente. Sin embargo, hay casos en que si se hace esto directamnte, pueden existir problemas y no obtenerse exactamente el efecto deseado. Ejemplos de esto son la aplicación de ecualización por histograma y la detección de orillas en imágenes a color.

4.6.1 Ecualización por histograma

Si aplicamos directamente la técnica de ecualización por histograma a una imagen representada en el modelo RGB, obtendríamos cambios de color (croma). Por esto se aplica usando el modelo HSI, sólo en la componente de intensidad (I), sin alterar las otros componentes (H y S). De esta forma se obtiene un mayor contraste sin modificar el "color" (croma) de la imagen original. Un ejemplo de mejora de contraste de una imagen utilizando esta técnica se presenta en la figura 4.18.

Figura 4.18: Transformación de una imagen de color mediante ecualización por histograma: (a) imagen original, (b) imagen ecualizada en intensidad.

4.6.2 Detección de orillas

En principio podemos aplicar las mismas técnicas que se utilizan en imágenes monocromáticas para detección de orillas en imágenes a color. Para esto se toma cada componente (R, G, B, por ejemplo) como una imagen monocromática y se aplica algún operador a cada una independientemente. Después se combinan todos las orillas detectadas (se considera normalmente el máximo o el promedio). Esto lo podemos hacer en los diferentes modelos.

RGB

En este caso se pueden presentar problemas ya que puede haber orillas que no impliquen un cambio fuerte en ninguna componente, pero si son notables en color o saturación. Un ejemplo de detección de orillas con este concepto se ilustra en la figura 4.19.

HSI

En principio los modelos perceptuales deben ser mejores ya que nosotros detectamos los cambios en estas componentes. Sin embargo, es difícil implementar la detección de orillas en croma por no ser lineal. Otra alternativa es definir técnicas especiales para detección de orillas en imágenes a color. Una técnica de este tipo se basa en el concepto de distancia de color entre pixels:

$$d = [(R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2]^{1/2,} \tag{4.20}$$

ó

$$d = [abs(R1 - R2) + abs(G1 - G2) + abs(B1 - B2)] \tag{4.21}$$

Se toma la distancia de cada pixel a sus vecinos (máscara de 3x3), se suman y se normalizan (dividir entre 8). De esta forma la "magnitud" de la orilla aumenta al aumentar la diferencia en intensidad, croma o saturación.

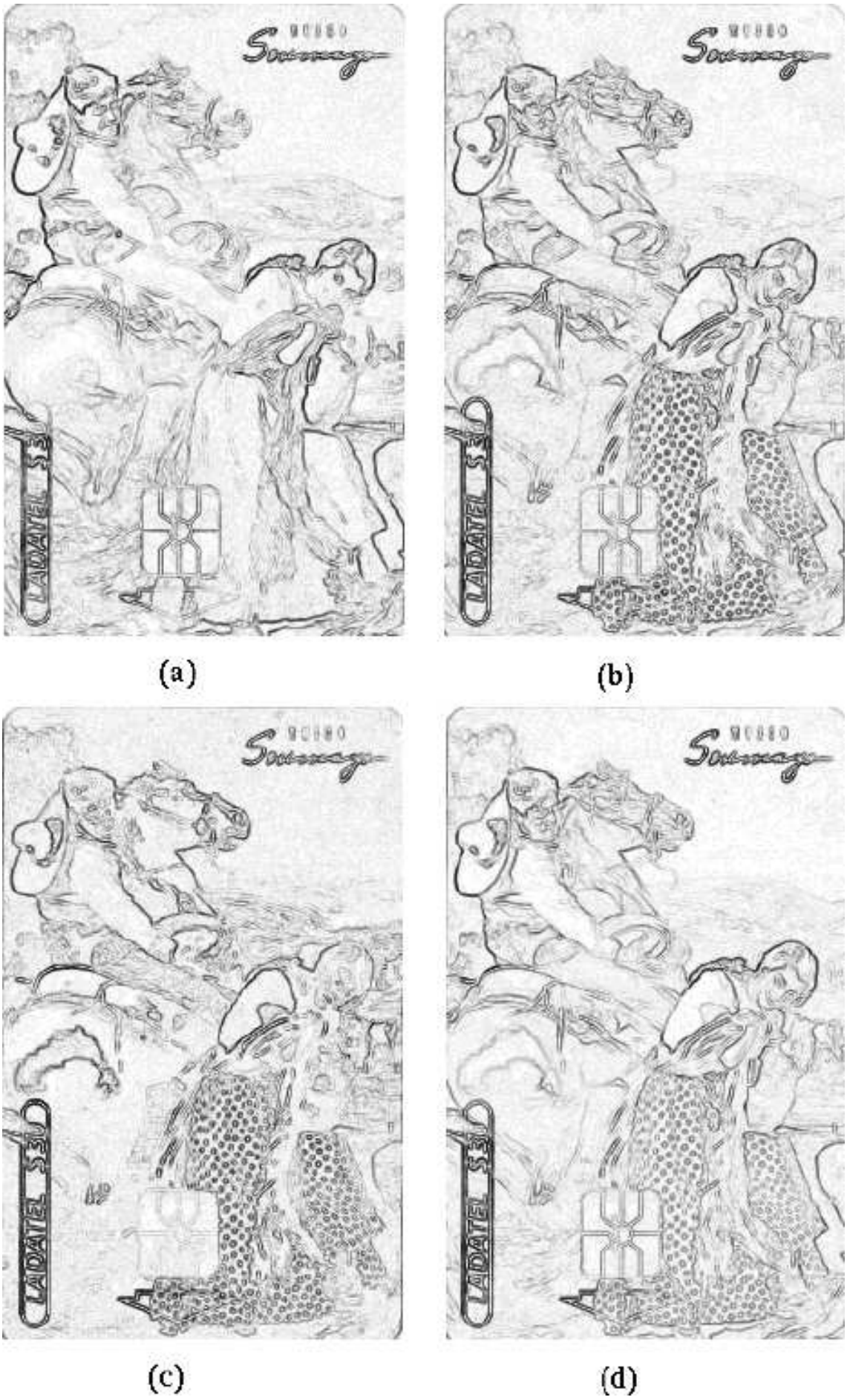


Figura 4.19: Ejemplo de detección de orillas con el operador Sobel: (a) Plano rojo. (b) Plano verde. (c) Plano azul. (d) Orillas sobre la imágen monocromática.

4.7 Referencias

El procesamiento de imágenes a color es relativamente reciente, por los altos requerimientos de memoria y cómputo requeridos. Kiver [62] trata a más profundidad los fundamentos de color. Referencias adicionales sobre los fundamentos y modelos de color se pueden consultar en libros de gráficas como el de Foley y Van Dam [23]. El libro de González y Woods [28] trata el uso de pseudo–color.

Menegos [79] realiza un análisis sobre la detección de orillas en imágenes a color. La aplicación de ecualización por histograma se comenta también en el libro de González y Woods [28]. Jones y otros [45] tratan la aplicación de modelos de color para la detección de piel en imágenes. La visión a color se ha estudiado en diferentes organismos, entre ellos en peces [66].

4.8 Problemas

1. ¿Se pueden tener diferentes colores primarios?, ¿Qué condiciones deben satisfacer?

2. ¿Cuáles son los 3 atributos *perceptuales* del color? Describe brevemente cada uno y muestra su interpretación gráfica en alguno de los modelos perceptuales de color.

3. ¿Qué es ecualizaci'on por histograma? ¿Qué efecto tiene en la imagen? ?Cómo se debe aplicar en una imagen a color y porqué?

4. Explica la diferencia entre los modelos "sensoriales" y "perceptuales" para representar el color. Da un ejemplo de c/u de estos tipos de modelos.

5. ¿Qu'e es un modelo perceptual del color? ¿C'omo se representa el color en este tipo de modelos? Muestra en forma gr'afica alguno de los modelos perceptuales incluyendo como se mide cada una de las componentes.

6. Demuestra geométricamente la relación entre HSI y RGB.

7. Muestra un ejemplo de una imagen sencilla (componentes R, G, B) en donde sea difícil detectar orillas en R, G, B y más fácil usando la técnica basada en distancia.

8. En capítulo se describe una técnica especial para detección de orillas en imágenes a color (RGB), la cual sólo detecta la magnitud pero no la dirección de la orilla. Propon una modificación para obtener también la dirección. Da la fórmula.

9. En cierta aplicación se tienen partes de 3 colores que se quieren diferenciar pero sólo se cuenta con una cámara monocromática. Propon una técnica para utilizar esta cámara para detectar los 3 diferentes colores.

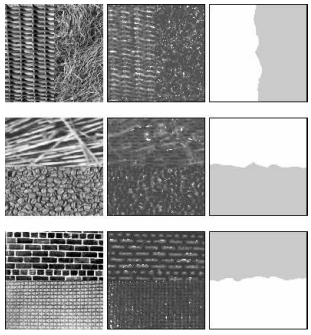
10. En aplicaciones de reconocimiento o seguimiento de personas, una forma incial de detección es utilizar color de piel. Propon una forma de diferenciar pixels de piel de otros pixels en una imágen, indicando el modelo de color que utilizarías y alguna forma de hacer la clasificación.

4.9 Proyectos

1. Implementar en en el laboratorio ecualización en color. Para ello primero convertir al modelo HSI, luego ecualizar en "I", y finalmente transformar a RGB y desplegar la imagen ecualizada.

2. Implementar en en el laboratorio una segmentación sencilla en base a color. Para ello obtener el histograma en R, G, y B de un tipo de objeto (por ejemplo caras de personas), obteniendo el rango de cada componente del objeto. Utilizar este rango para luego "separar" objetos similares en imágenes, cuando estén dentro del rango de cada componente.

3. Repetir el problema anterior, utilizando “H” (del modelo HSI) en lugar de las componentes RGB. Comparar los resultados.



Capítulo 5

Tratamiento de texturas

5.1 Introducción

Muchos objetos o regiones no son uniformes, sino están *compuestas de pequeños elementos indistinguibles y entrelazados* que en general se conoce como "textura". La figura 5.1 muestra ejemplos de diferentes tipos de texturas. Para algunas de ellas los elementos básicos o primitivos son claramente distinguibles, como el caso de los ejemplos de las texturas de frijoles, ladrillos y monedas. Para los otros ejemplos es más difícil definir los elementos primitivos.

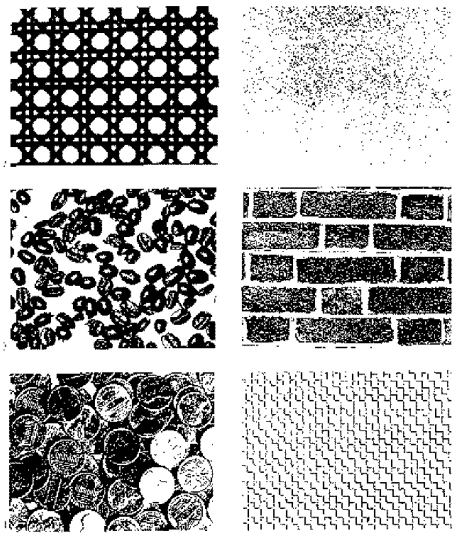


Figura 5.1: Ejemplos de texturas: bejuco, papel, frijoles, ladrillo, monedas, trenza de alambre (de arriba a abajo, de ezq. a derecha).

La textura en una imagen tiene que ver mucho con la resolución. Lo que a cierta resolución son objetos claramente distinguibles, a una resolución menor se ve como cierta textura y una resolución aún menor puede parecer una región uniforme.

El analizar y reconocer diferentes tipos de textura es útil para el reconocimiento de ciertas clases de objetos e incluso en otros aspectos de visión como la determinación de forma tridimensional (*shape from texture*). Existen diferentes formas de describir los tipos de textura, que se clasifican en:

- modelos estructurales,
- modelos estadísticos,

- modelos espectrales.

Veremos primero el concepto de elementos o primitivas de textura y después cada uno de los tipos de modelos para describir texturas.

5.2 Primitivas de las texturas

A los elementos básicos o primitivas de textura se les denomina *texel* (*texture element*). Podemos definir un *texel* como "una primitiva visual con ciertas propiedades invariantes que ocurre repetidamente a diferentes posiciones, deformaciones y orientaciones en un área" Ejemplos de *texels* se ilustran en la figura 5.2.

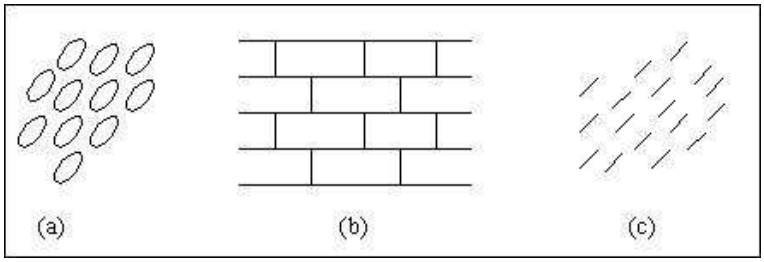


Figura 5.2: Ejemplos de *texels* o elementos constituyentes de las texturas: (a) elipses, (b) rectángulos, (c) segmentos de línea.

Las propiedades invariantes de los *texels* pueden ser:

- forma,
- tamaño,
- nivel de gris,
- color.

Es importante conocer el número de *texels* en cierta área, aunque es computacionalmente difícil calcularlo. Un número muy pequeño de texels haría que se pudieran distiguirse como objetos aislados; en tanto que un número muy grande puede hacer que visualmente veamos la superficie "global" uniforme. Por lo tanto el número de texels tiene que ver con la resolución. Las texturas pueden ser jerárquicas, observándose un tipo de textura a cierta resolución y otra textura a resoluciones mayores o menores. Por ejemplo, una pared de ladrillo tiene cierta textura si la observamos desde lejos (rectángulos) y otra textura diferente si la observamos de muy cerca (textura de los ladrillos).

Algunas texturas pueden ser completamnte caracterizadas en dos (2D) dimensiones, mientras que para otras se requiere un modelo en tres dimensiones (3D). Para las texturas caracterizables en 2D, los *texels* pueden ser descritos a nivel imagen, como curvas o regiones en 2D. Tal es el caso de los elementos en la figura 5.2 o de los ejemplos de texturas de frijoles y ladrillos. Los elemenos en primitivos de texturas en 3D, requieren caracterizarse con modelos tridimensionales, como es el caso del ejemplo de la textura de monedas. Como se puede observar en algunos de las texturas en la figura 5.1, es difícil definir un elemento básico o *texel* para algunos tipos de texturas. En estso casos las textura se caracterizan de manera estadística, como veremos más adelante.

Las texturas para las que se pueden identificar los *texels* constitutivos, se pueden caracterizar en base a dichos elementos en base a *modelos estructurales*.

5.3 Modelos Estructurales

Las texturas altamente regulares se pueden describir en términos de elementos (polígonos) que en pocas formas básicas se repiten uniformemente en la superficie.

Las texturas regulares son aquellas en que cada polígono tiene el mismo número de lados. Existen tres texturas regulares para un plano, como se ilustra en la figura 5.3.

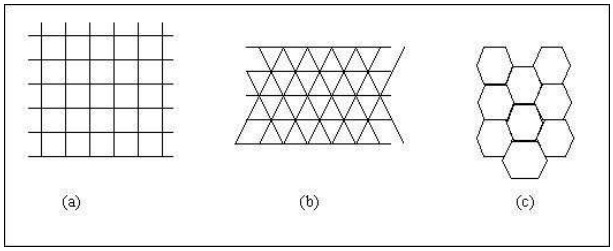


Figura 5.3: Texturas regulares: (a) elemento rectangular, (b) elemento triangular, (c) elemento hexagonal.

Las texturas semi–regulares están formadas por dos tipos de polígonos con diferente número de lados. Hay seis tipos de texturas semi-regulares para un plano que se muestran en la figura 5.4. Podemos describir este tipo de estructuras, regulares y semi-regulares, en forma muy compacta mediante el número de lados de los polígonos adyacentes a un vértice. Para ello se identifican en froma secuencial los polígonos alrededor de un vértice. Para cado uno se obtiene el número de lados, y estos números se concatenan formando un código que distingue a la textura. Por ejemplo:

- Textura regular hexagonal: (6,6,6).
- Textura semi-regular triangular-hexagonal: (3,6,3,6).

No sólo es importante la estructura que indica la forma de los elementos sino también la que nos da su posicionamiento en el plano. Esta se obtiene uniendo los centros de cada uno de los polígonos. De esta forma obtenemos una nueva "textura" que se le conoce como la *dual*. La figura 5.5 ilustra el posicionamiento de los polígonos para la textura hexagonal y la textura dual que corresponde a la triangular. Sucede lo contrario al invertir los papeles; es decir, la textura hexagonal es la dual de la triangular.

Una forma más general y poderosa de describir a las texturas estructuradas es mediante modelos gramaticales.

5.3.1 Modelos gramaticales

Otra forma de describir texturas regulares es mediante un conjunto de formas básicas y reglas sencillas para combinarlas. Podemos pensar en éstas formas básicas como símbolos y con ellos describir texturas mediante gramáticas.

Por ejemplo, la gramática:

- símbolo: $\triangle$,
- regla: $S \rightarrow \triangle S$,

puede generar patrones de la forma:

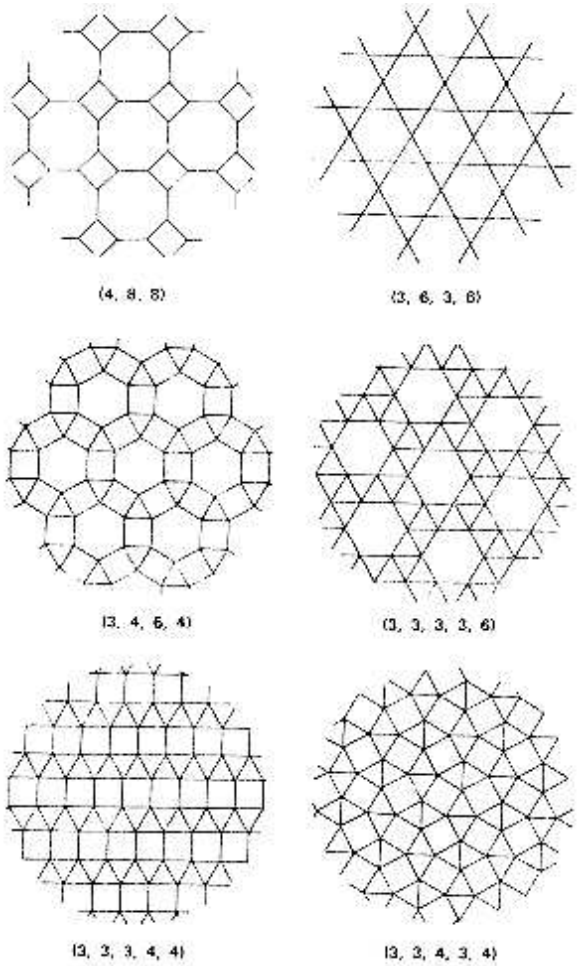


Figura 5.4: Ejemplos de texturas semi-regulares. Para cada una se indica su codificación en función de los polígonos en un vértice.

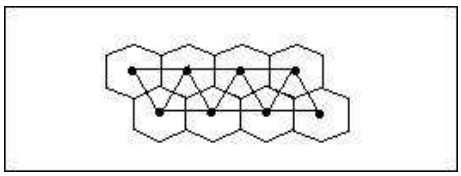


Figura 5.5: División de posicionamiento de texels para la textura hexagonal. En este caso la textura dual es triangular.

△△△△△△△△△△△

Que corresponde a un textura uniforme en base a triángulos (en una dimensión). Formalmente, una gramática de forma se define como el tuple V_t, V_m, R, S, donde cada elemento es a su vez un conjunto que se define de la siguiente manera:

1. Un conjunto finito de formas V_t (elementos terminales).
2. Un conjunto finito de formas V_m tal que $V_i \cap V_m = \emptyset$ (elementos de marca).
3. Un conjunto R de pares ordenados (u, v) donde u es una forma que consiste de un elemento de V_t^+ y V_m^+, y v de un elemnto de V_t^* y V_m^* (reglas).

4. Una forma S que consiste de elementos de V_t^* combinados con elementos de V_m^* (forma inicial).

V^+ se forma de un conjunto finito de elementos de V, donde cada elemento se puede usar varias veces a diferentes orientaciones, posiciones y escalas; V^* es V^+ unión la forma vacía. En la figura 5.6 se muestra un ejemplo de una gramática para definir la textura hexagonal.

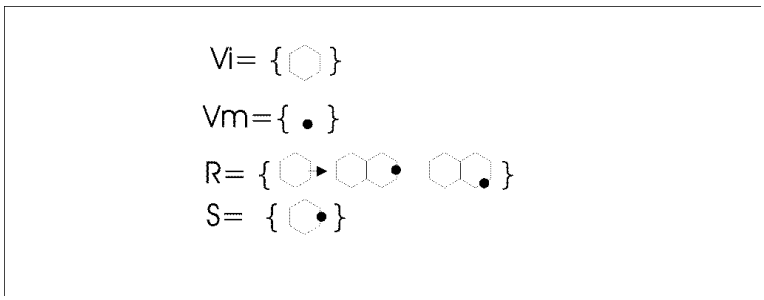


Figura 5.6: Gramática para la textura hexagonal.

Una gramática de textura se puede utilizar en dos sentidos:

1. Generación. Se pueden aplicar las reglas (R) para generar texturas de acuerdo a la gramática. Esto puede utilizarse en graficación por computadora.

2. Reconocimiento. Se aplican las reglas en sentido *inverso*, hasta que se obtenga una forma incial, S. Si se llega a una de las formas inciales se ha reconocido como el tipo de textura representada por la gramática; de otra forma no se reconoce como de ese tipo.

En base a la utilización como reconocimiento, los modelos gramaticales se aplican a distinguir diferentes clases de texturas regulares en imágenes. Este concepto se puede extender a otras gramáticas, un poco más complejas, como son las gramáticas de árboles y de arreglos.

Para otro tipo de texturas no regulares se utilizan otro tipo de modelos como son los estadísticos y los de energía espacial.

5.4 Modelos Estadísticos

Muchas texturas no tienen una estructura tan regular y uniforme, por lo que es más adecuado describirlas en términos de modelos estadísticos. Para esto se utilizan técnicas de reconocimiento estadístico de patrones. Un primer método, relativamente simple, es utilizar el histograma de niveles de gris y caracterizarlo mediante sus momentos. En la figura 5.7 se muestran ejemplos de diferentes texturas no regulares y su correspondiente histograma de intensidades.

El momento n respecto a la media m se define como:

$$\mu_n(z) = \sum_i (z_i - m)^n P(z_i)$$

Donde z_i es el nivel de gris y $P(z_i)$ es su respectiva probabilidad, estimada a partir del histograma.

El segundo momento o varianza es particularmente útil ya que nos da una medida de la uniformidad o suavidad de la región. Si definimos una medida auxiliar en términos de la varianza (σ^n):

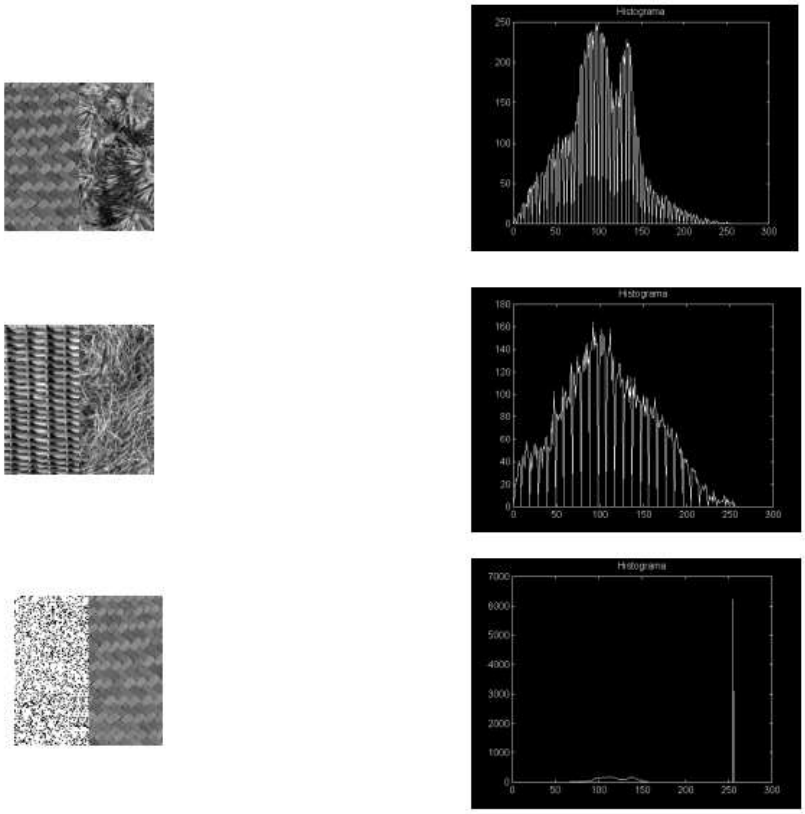


Figura 5.7: Ejemplos de texturas no regulares y sus histogramas. Del lado izquierdo se muestran 3 ejemplos de mosaicos con dos texturas diferentes cada uno, del lado derecho se ilustra el histograma correspondiente a cada imagen. Se puede notar que en dos casos, primero y tercero, se distinguen dos "picos" en el histograma, correspondientes a cada textura.

$$R = 1 - \frac{1}{(1 + \sigma^n(z))}$$

Entonces R es 0 para áreas de intensidad uniforme y se aproxima a 1 para áreas de muy alta varianza. Se pueden utilizar momentos mayores, en particular el tercero (medida de desplazamiento) y cuarto (medida de uniformidad relativa). También se pueden utilizar momentos de orden mayor, pero estos ya no tienen una interpretación intuitiva. En conjunto proveen información para la discriminación de diferentes texturas. En la figura 5.8 se ilustran en forma cualitativa diferentes distribuciones (histogramas) que varían en los diferentes momentos, del primero al cuarto.

Los diferentes momentos se pueden agrupar en un vector, lo que nos da un vector de características (*feature vector*) de la textura correspondiente:

$$V = (v_1, v_2, ..., v_n)$$

Donde n es el número de momentos utilizados. Este vector condensa la descripción de la información relevante de la textura en pocos parámetros. Entonces la textura se pueden "ver" como un vector en un espacio *n–dimensional*. En la tabla 5.1 se muestran los vectores de características (primeros 3 momentos) obtenidos para los histogramas de los ejemplos de texturas de la figura 5.7.

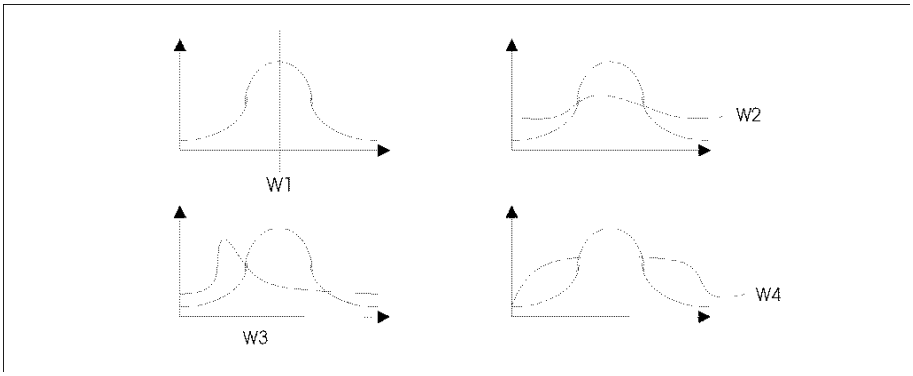


Figura 5.8: Ilustración de las diferencias cualitativas de histogramas para los primeros cuatro momentos. En (a) se ilustran dos distribuciones que difieren en el primer momento (promedio), en (b) que difieren en el segundo momento (varianza), en (c) que difieren en el tercer momento (desplazamiento), y en (d) que difieren en el cuarto momento (uniformidad relativa)

Tabla 5.1: Momentos para Ejemplos de Texturas.

Imagen	Momento 1	Momento 2	Momento 3
1	101.98	1594.7	-33.33 $\times 10^9$
2	109.16	2667.6	-1.67 $\times 10^{10}$
3	155.33	4717.8	-8.24 $\times 10^{10}$

En general, en reconocimeinto de patrones se busca describir a cada patrón como un vector o región en el espacio *n–dimensional*. El proceso de reconocimiento consiste en encontrar, para un patrón desconocido, el vector "más" cercano que corresponde a la clase que describe dicho ejemplo. Esta idea se decribe graficámente en la figura 5.9, considerando en este caso dos características: X_1 y X_2. Cada punto en la la figura representa una imagen de una textura; y cada grupo de puntos diferentes en la figura representa una clase, en este caso un tipo de textura.

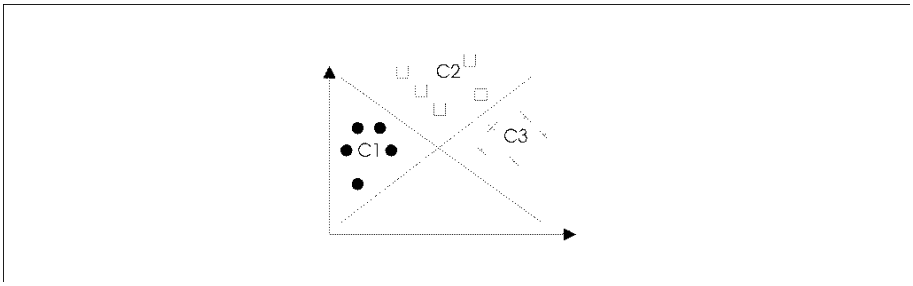


Figura 5.9: Representación gráfica de vectores de características para 2 momentos. Cada tipo de punto (cuadrado, tríangulo, círculo) represnta un tipo de textura.

Las características o atributos deben ser seleccionados de forma que exista cierta correlación entre elementos de la misma clase; es decir, que forman grupos o *clusters* en el espacio *n-dimensional*. El ejemplo de la figura 5.9 es un cado ideal, en el que las clases se pueden separar facilmente (en este caso por líneas rectas). En general, no es tan sencillo lograr esta separación.

Existen varias formas de asignar un elemento a una clase específica. Un alternativa es usar la distancia (d) euclidiana, asignando el elemento (textura) desconocido a la clase con d mínima. Para ello se obtiene el "centro de masa" de cada clase (w_i) y se calcula la distancia euclidiana del vector desconocido (v) a cada centro, seleccionando la clase con dictancia mínima. Esto es:

$$Clase(V) = j, \tag{5.1}$$

donde:

$$d(j) = min[d(v, w_i)], \forall_i. \tag{5.2}$$

Esta forma de clasificación corresponde a lo que se como el *vecino más cercano*. Existen otras técnicas de clasificación, como el clasificadore bayesiano, redes neuronales y redes bayesianas, que se verán en los capítulos de visión de alto nivel.

El aspecto más importante en este tipo de técnicas es las selección de atributos. Diferentes alternativas de transformaciones se han desarrollado para la clasificación de texturas, entre estas se encuentran:

- momentos,
- energía en el dominio espacial,
- matrices de dependencia espacial,
- transformada de Fourier.

Hasta ahora hemos presentado sólo la representación en base a momentos, en las siguientes secciones se presentan las otras técnicas.

5.4.1 Energía en el dominio espacial

La técnica de energía en el dominio espacial consiste en hacer una transformación de la imagen para obtener lo que se denomina una "transformada de energía de textura" que es en cierta forma análoga al espectro de potencia de Fourier. Para ello, se aplica el siguiente procedimiento:

1. Ecualización por histograma de la imagen.
2. Convolución con 12 funciones base (máscaras - $h_1, ...h_{12}$) para obtener 12 nuevas imágenes: $f' = f * h_k$
3. Obtención del promedio absoluto de una ventana de 15x15 y su substitución por cada pixel central de la ventana: $f'' = \sum \mid f' \mid$
4. Clasificación de cada pixel de acuerdo al vecino más cercano (distancia mínima) respecto a las 12 imágenes obtenidas (atributos).

Esta técnica ha sido aplicada exitosamenta para clasificación de texturas. El aspecto clave es la selección de las máscaras. La figura 5.10 muestra un ejemplo de una de las máscaras que se han utilizado y que han dado mejores resultados.

$$\begin{vmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{vmatrix}$$

Figura 5.10: Ejemplo de función base (máscara) utilizada para la clasificación de texturas.

5.4.2 Matrices de dependencia espacial

Para el método de matrices de dependencia espacial se obtiene una matriz intermedia de medidas a partir de la imagen y de esta matriz se definen las características o atributos para clasificación. El procedimiento se puede dividir en las siguientes etapas:

1. Obtener la matrices intermedias $S(d,\theta)$, de forma que $S(i,j \mid d,\theta)$ es el número de veces que le nivel de gris i esta orientado respecto al j de forma que:

$$f(x) = i, \quad f(y) = j, \quad y = x + (d \times cos(\theta), d \times sen(\theta))$$

   Cada matriz S es una matriz de 2 dimensiones, $M \times M$, donde M es el número de niveles de gris de la imagen. Es decir, cada elemento i,j de la matriz indica el número de veces que un pixel de valor i tiene una relación de (d,θ) respecto al pixel de valor j.

2. Obtener características de la matriz S. Para ello se normaliza (divide entre el número de pares de pixels) S y se obtiene la matriz P, a partir de la cual se pueden calcular los siguientes atributos:

   - Energía: $\sum_i \sum_j P_{ij}^2$
   - Entropía: $\sum_i \sum_j P_{ij} log P_{ij}$
   - Correlación: $\sum_i \sum_j (i - m_x)(j - m_y) P_{ij}$
   - Inercia: $\sum_i \sum_j (i-j)^2 P_{ij}$
   - Homogeneidad local: $\sum_i \sum_j \frac{1}{1+(i-j)^2} P_{ij}$

   Donde m_x es la media en x y m_y es la media en y.

3. Realizar la clasificación de texturas en base a las características obtenidas.

Normalemente se tienen pocos valores de (d,θ). Por ejemplo, podrían utilizarse 2 valores de d: 1 y 2; y 4 valores de θ: 0, 45, 90 y 135 grados. Esto daría 8 matrices S, de forma que se pueden obtener los atributos para cada una. La figura 5.11 ilustra como se obtiene la relación entre pixels para formar las matrices S

0	0	0	0	0
0	0	0	15	0
0	0	5	0	10
0	0	0	0	0
0	0	0	0	0

Figura 5.11: Ejemplo de la obtención de la matriz intermedia S. Para el pixel con valor 5, se obtiene una relacion $d = 1$ y $\theta = 45$ con el pixel valor 15, lo que implica aumentar en 1 la posición [5,10] de la matriz $S(1, 45)$. El pixel 5 tiene una relación $(2, 0)$ con el pixel 10, etc.

5.5 Modelos Espectrales

La transformada de Fourier es adecuada en describir información "global" en la imagen, en especial patrones periódicos. Este es el caso de las texturas, generalmente, por lo que los modelos espectrales proveen buenas características para su descripción y clasificación. En particular, hay 3 características del espectro que son adecuadas para la descripción de texturas:

1. La amplitud de los picos prominentes dan la dirección principal de los patrones en la textura.

2. La localización de los picos en frecuencia indican el periodo espacial de los patrones.
3. Eliminando componentes periódicas mediante filtros en Fourier, se pueden dejar sólo las componentes a–periódicas a las que se les aplica técnicas estadísticas.

Estas características son más fáciles de detectar convirtiendo el espectro a coordenadas polares (fig. 5.12):

$$F(u,v) \rightarrow S(r,\theta) \tag{5.3}$$

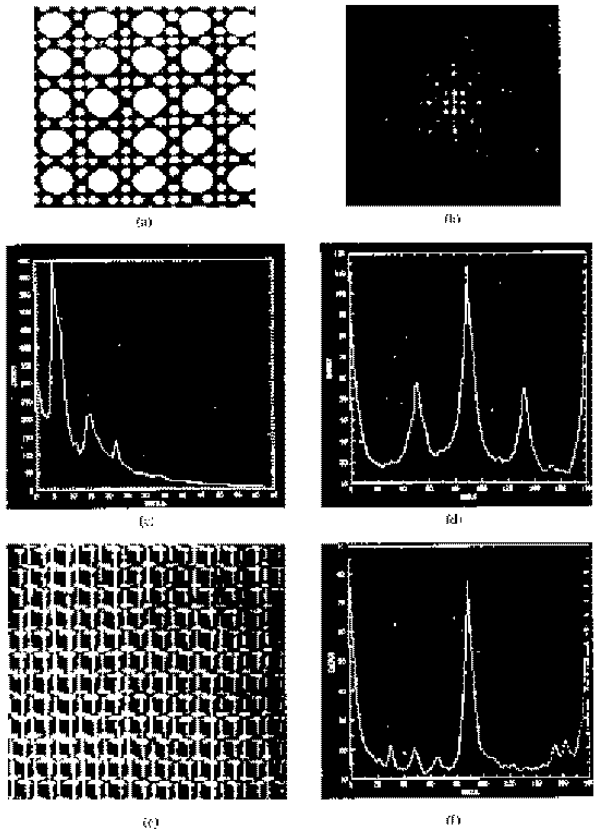


Figura 5.12: Ejemplos de espectros en coordenadas polares para diferentes texturas periódicas: (a) imagen de una textura, (b) espectro, (c) gráfica del espectro en r (radio), (c) gráfica del espectro en θ (ángulo), (d) imagen de otra textura, (e) gráfica del espectro en θ (ángulo).

Una descripción global se puede obtener integrado (sumando en el caso discreto) la transformada en una dimensión (con la otra variable constante):

$$S(r) = \sum_{\theta=0}^{\pi} (S_\theta(r)), \tag{5.4}$$

$$S(\theta) = \sum_{r=0}^{R} (S_r(\theta)) \tag{5.5}$$

Considerando que se discertiza el radio en R diferentes valores y el ángulo en Q diferentes valores, se obtiene un vector de $R+Q$ características que describen en base a la energía espectral a la textura. Estas características se pueden utilizar como entrada a un clasificador para descriminar diferentes tipos de texturas.

La figura 5.12 muestra dos imágenes de texturas periódicas y sus espectros de Fourier correspondientes. En este ejemplo, se puede ver la diferencia en la componente angular del espectro (figura 5.12 (d) y (f)) para las dos texturas.

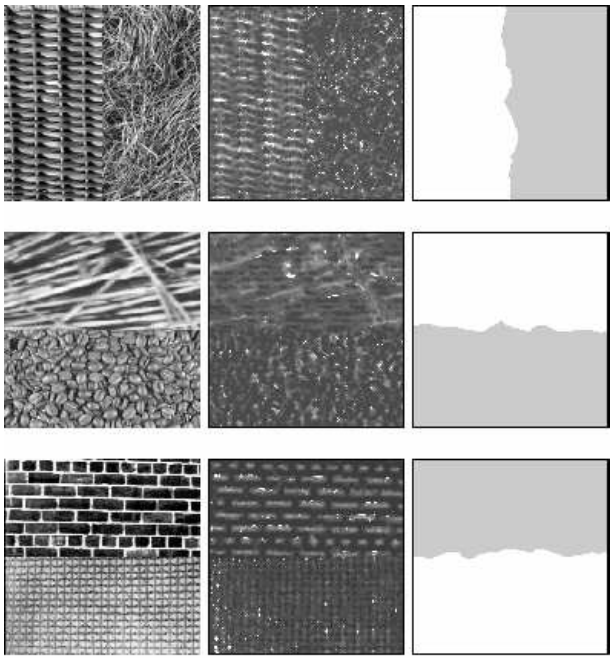


Figura 5.13: Ejemplos de segmentación de texturas. Se ilustran 3 ejemplos de mosaicos con dos texturas cada uno. Del lado derecho se tiene la imagen original, en la parte media los atributos de escala de las texturas, y en la parte derecha la separación de las texturas con un nivel de gris diferente para cada clase.

5.6 Aplicaciones

El análisis de texturas tiene diversas aplicaciones en procesamiento de imágenes y visión computacional. Entre las principales aplicaciones se encuentran:

1. Segmentación de imágenes. Una forma de dividir una imagen en diferentes partes es mediante su separación en base a texturas.
2. Reconocimiento de objetos. Diferentes clases de objetos pueden ser distinguidos en base a su textura, por ejemplo la clasificación de diferentes tipos de cultivo en imágenes aéreas.
3. Forma a partir de textura. Se puede utilizar información de como se deforman los *texels* en la imagen para inferir la orientación de la superficie, como una ayuda a recuperar la tercera dimensión.

La figura 5.13 ilustra un ejemplo de segmentación de texturas en base a características obtenidas con filtros gaussianos a diferentes escalas (multi-escala). A estos atributos de escala (parte intermedia de la figura) se les aplicó un proceso de regularización para realizar la segmentación de texturas (para mayor información, ver la sección de referencias y el capítulo de segmentación).

En este capítulo hemos visto, fundamentalemnte, diversas técnicas para describir una textura. Esta descripción se combina con técnicas de clasificación para la segmentación y reconocimiento en base a texturas. Las técnicas de reconocimiento o clasificación se verán más en detalle en los capítulos de visión de nivel alto. En el capítulo siguiente se analizará el uso de textura para obtención de forma.

5.7 Referencias

El análisis de texturas es una área que se ha desarrollado desde hace tiempo y en la cual continua investigándose. La mayor parte de los trabajos se han enfocado a la caracterización y segmentación de texturas. Otros libros presentan un resumen del área, en particular el de Ballard y Brown en el Capítulo 6.

Julesz ha publicado los trabajos clasicos sobre la percepción de texturas [46, 47], así como los elementos primitivos o *texels* [48]. Los modelos en base a gramática para texturas se introdujeron en el área a partir de los trabajos de K.S. Fu [72, 73]. Haralick [**?**] analiza los métodos de matrices de dependencia espacial. El enfoque basado en modelos espectrales se describe en [4].

Una colección importante de diferentes texturas, que tradicionalmente se ha utilizado para probar algoritmos, es la de Brodatz [8].

El área de reconocimiento de patrones ha sido extensamente estudiada. Los libros de Duda & Hart [19] y de Tou & González [121] presentan una introducción general a este campo. El libro de Tomita et al. [118] proveen una revision en detalle de tratamientos estadisticos y estructurales. Tambien vease [95] para una comparacion técnicas no vistas en las anteriores referencias, e.g. wavelets, filtros de cuadratura, filtros de Gabor (tambien en [35]), etc.

Otros trabajos que difícilmente entran en las categorias anteriores son los basados en dimensiones fractales que se describen en [86], así como el uso de campos aleatorios de Markov [12, 13] y de filtros multiescala [27].

5.8 Problemas

1. ¿Qué es textura? ¿En qué puede servir el analizar la textura de una imagen?

2. ¿Qué tipos de modelos de texturas hay y a que tipos de texturas están orientados?

3. ¿Cómo se puede obtener un vector de caracterízticas que describa una textura? Dado este vector, ¿Cómo se puede utilizar para clasificar texturas?

4. Determina la división dual (posicionamiento) para la textura semi-regular (4,8,8).

5. Obten una gramática de forma para la textura semi-regular (4,8,8).

6. Algunos objetos pueden presentar diferentes texturas dependiendo de la distancia, como el caso de la pared de ladrillos o pasto. Decribe, en forma muy general, una técnica para evitar este problema, de forma que pudiera reconocerse la superficie a diferentes distancias.

7. Consider la textura semi-regular descrita por el código "(3, 6, 3, 6)". (a) Dibuja dicha textura. (b) Obten su división dual, dibujala y da su código. (c) Especifica una gramática de forma para esta textura (la original).

8. Para cada una las texturas semi–regulares de la figura 5.4: (a) Da el código correspondiente. (b) Dibuja la textura dual y también indica su código. (c) Describe una gramática de forma para esta textura.

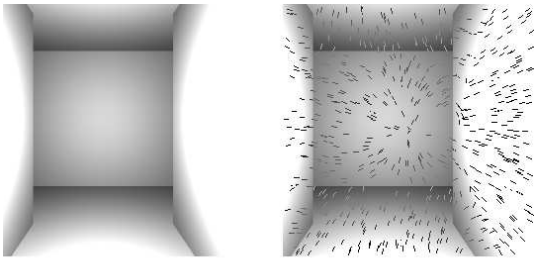
9. Dado el espectro en radio y ángulo (r y θ) de diferentes texturas, plantea una forma de utilizar dichos espectros para diferenciar diferentes tipos de texturas.

10. Una aplicación de texturas es para segmentar imágenes que tengan diferentes texturas. Considerando quue tengas un método que distinga diferentes texturas en una "pequeña" ventana de pixels, ¿Cómo utilizarías dicho método para separa diferentes regiones en una imagen en base a texturas? ¿Qué problemas podrían presentarse al realizar la separación?

5.9 Proyectos

1. Implementar en en el laboratorio el análisis de texturas en base a su histograma. Para ello: (a) obtener el histograma de una imagen de textura, (b) obtener los primeros 4 momentos del histograma, (c) probar con diferentes imágenes de texturas y comparar los momentos obtenidos.

2. Implementar en en el laboratorio la clasificación de texturas en base a matrices de dependencia espacial. Para ello obtener la matriz en una ventana de la imagen, considerando una discretización en 8 niveles de gris, 8 direcciones y una distancia máxima de $n/2$, donde n es el tamaño de la ventana. Luego calcular los atributos globales de la matriz. Probar con diferentes imágenes de texturas y comparar los atributos obtenidos.



(a) (b)

Capítulo 6

Visión tridimensional

6.1 Introducción

El proceso de proyección de un objeto en el mundo tridimensional (3-D) a una imagen bidimensional (2-D) no es un proceso reversible. Se pierde información en esta transformación, ya que una línea en 3-D se convierte en un punto en la imagen, por lo que no es invertible en el sentido matemático. Existen, en principio, un número infinito de escenas que pueden resultar en la misma imagen, como se ilustra en forma simplificada en la figura 6.1.

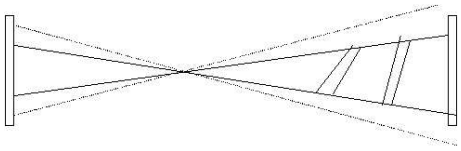


Figura 6.1: Proyección: 3D a 2D. Diferentes objetos en el mundo (3D) generan la misma proyección en la imagen (2D).

Sin embargo, existen alternativas para poder recuperar la tercera dimensión que se ha perdido en el proceso de proyección. Una alternativa es usar dos imágenes mediante visión estereoscópica. Otras consisten en utilizar otras propiedades de la imagen como sombreado o textura para obtener un estimado de la profundidad, o al menos de la profundidad relativa (gradiente).

En las siguientes secciones veremos 3 de estos enfoques para obtener información de 3D o profundidad: estereo, forma de sombreado y forma de textura. Otra alternativa es utilizar información de una secuencia de imágenes (forma de movimiento), que comentaremos en el capítulo de movimiento.

6.2 Visión estereoscópica

Una forma de recuperar la tercera dimensión es utilizar dos (o más) imágenes, en analogía con los sistemas de visión biológicos. Se colocan dos cámaras en posiciones distintas a una distancia conocida para obtener dos imágenes de cada punto de donde se puede recuperar su posición en 3-D (ver figura 6.2). El algoritmo básico consiste de cuatro etapas:

1. Obtener dos imágenes separadas por una distancia d.
2. Identificar puntos correspondientes.
3. Utilizar triangulación para determinar las dos líneas en $3-D$ en las que esta el punto.
4. Intersectar las líneas para obtener el punto en $3-D$.

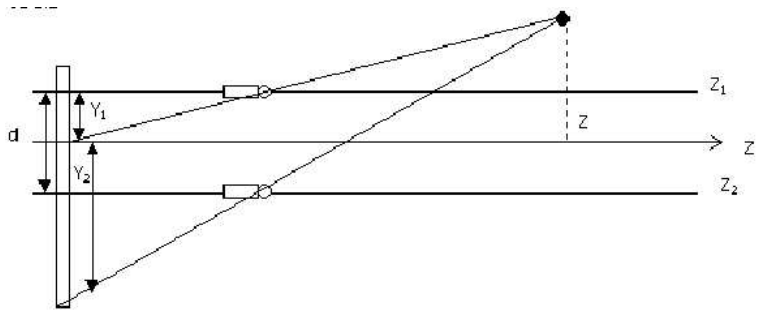


Figura 6.2: Visión estereoscópica. Un punto (z) tiene dos proyecciones diferentes en las cámaras, $y1, y2$. Las cámaras están separadas por una distancia d.

Una forma sencilla de resolver el problema de geometría (pasos 3 y 4 del algoritmo) es considerando dos cḿaras colineales (generalmente sobre el mismo eje horizontal). Con dos cámaras colineales separadas una distancia conocida $2d$, tendremos dos imágenes de cada punto (X, Y). Entonces, las ecuaciones para la proyección perspectiva del modelo geométrico para dos cámaras son las siguientes:

$$y' = \frac{(Y - d)f}{(f - Z)} \tag{6.1}$$

$$y'' = \frac{(Y + d)f}{(f - Z)} \tag{6.2}$$

De donde podemos obtener el valor de Z:

$$Z = \frac{f - 2df}{(y' - y'')} \tag{6.3}$$

De aquí podríamos pensar que el extraer información de profundidad es aparentemente simple teniendo un sistema con dos cámaras (estereo).

El problema principal es el segundo paso del algoritmo básico, ya que no es fácil identificar los puntos correspondientes entre las imágenes. Una alternativa es usar correlación o *template matching*, otra es una algoritmo de relajación. En las siguientes secciones se describen ambos enfoques.

6.2.1 Correlación

El enfoque de correlación consiste en tomar una pequeña porción de una imagen (*template*), y convolucionarlo con la otra imagen para encontrar la sección que de una mayor correlación, indicando la posible localización de esa característica, y cálculando de esta forma su distancia. Este enfoque se muestra en forma gráfica en la figura 6.3, en donde la "esquina" de un objeto de la imagen 1 se "busca" en la imagen 2. El proceso de búsqueda consiste en hacer una convolución del patrón (*template*) con la segunda imagen (en forma análoga al filtrado en el dominio espacial), estimando la correlación o similitud; y seleccionando el área de la imagen de mayor correlación.

Existen diferentes formas de estimar la similitud, las dos más comunes son las siguientes. Una es mediante el cálculo de la correlación cruzada:

$$\sum_{i=0}^{N-1} \sum_{j=1}^{M-1} T(i, j)I(i, j) \tag{6.4}$$

Otro es mediante la suma de diferencias cuadráticas:

$$\sum_{i=0}^{N-1}\sum_{j=1}^{M-1} -[T(i,j) - I(i,j)]^2 \tag{6.5}$$

Donde $T(i,j)$ son los valores de la ventana o *template* e $I(i,j)$ corresponde a la sección de la imagen, considerando una ventana de $N \times M$. La correlación cruzada tiende a dar mayor *peso* a los pixels de mayor intensidad, por lo que en general se prefiere la suma de diferencias o error cuadrático.

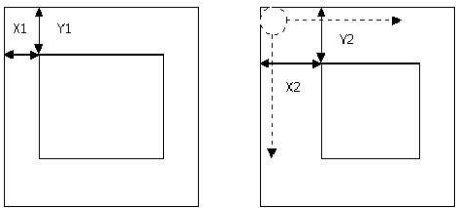


Figura 6.3: Correlación. Una región (ventana) de la imagen izquierda se convoluciona con la imagen derecha, hasta encontar la localidad de mayor simulitud.

Aunque éste método es bastante simple, tiene varios problemas importantes:

- La complejidad computacional del proceso de convolución.
- Las imágenes pueden verse diferentes desde dos puntos de vista, por lo que una mayor correlación no es confiable.

Una alternativa es utilizar objetos cracterísticos para hacer la correlación (esquinas, orillas, etc.). Este enfoque se conoce como basado en características (*feature based*) y también es común en visión estero, ya que se reduce la complejidad computacional al no considerar toda la imagen.

6.2.2 Relajación

El método de *relajación* se basa en la observación de que los humanos podemos percibir profundidad (3-D) de imágenes que no tienen atributos u objetos conocidos, como lo son lo que se conoce como "estereogramas de puntos aleatorios" (ver figura 6.4).

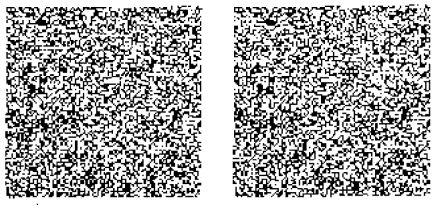


Figura 6.4: Estereograma de puntos aleatorios. Una sección de la imagen de la izquierda esta desplazada en la imagen de la derecha lo que da el efecto de diferentes profundidades.

Si tenemos una serie de puntos en el espacio y los proyectamos en dos imágenes, existen muchas posibles configuraciones consistentes con ambas imágenes, de las cuales sólo una es la correcta. Este fenómeno, el cual se ilustra en la figura 6.5, se conoce como el problema de la *correspondencia* de puntos en estereo.

Sin embargo, hay dos principios que podemos aplicar para obtener la solución más probable:

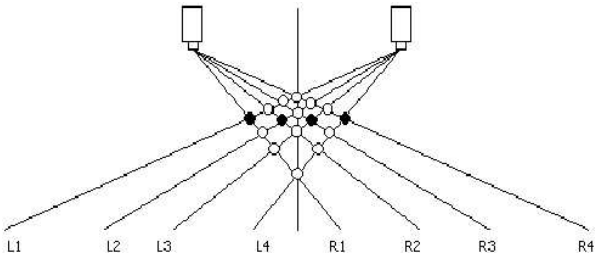


Figura 6.5: El problema de la *correspondencia* de puntos estereo. Cada uno de los puntos de la vista izquierda (L) puede aparear cualquiera de lso puntos de la vista derecha (R). De los 16 posibles apreamientos, sólo 4 son correctos (círculos rellenos), los otros 12 son falsos.

- Cada punto en la imagen sólo tiene un valor de profundidad.
- Los puntos tienen valores de profundidad cercanos a sus vecinos.

Estos dos principios se traducen en 3 reglas:

1. **Compatibilidad**: sólo elementos similares tienen correspondencia entre las imágenes (puntos negros con puntos negros).
2. **Únicos**: un elemento de una imagen sólo corresponde a un elemento de la otra (un punto negro con sólo otro).
3. **Continuidad**: la distancia entre puntos correspondientes varíá suavemente en casí todo la imagen.

En base a estas reglas se establece un algoritmo de relajación para obtener la disparidad a partir de dos imágenes de puntos. Para ello se considera una matriz $C(x, y, d)$, donde x, y corresponde a los diferentes puntos de la imagen y d a la disparidad entre éstos. La matriz tiene un uno para cada (x, y, d) que indique una correspondencia, y un cero en los demás. En la figura 6.6 se muestra la matriz de disparidad vs. coordenadas que es la base del algoritmo de relajación.

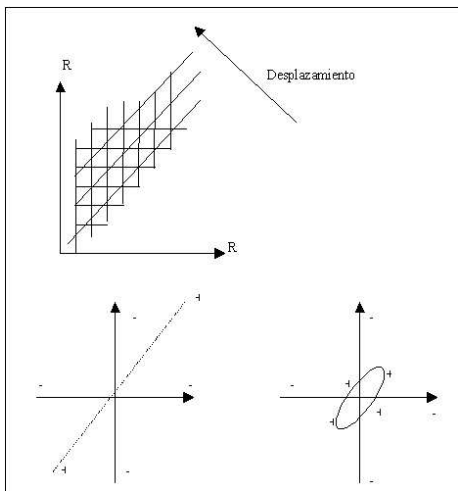


Figura 6.6: Algoritmo cooperativo de Marr.

Entonces el algoritmo relajación para estereo se presenta a continuación.

Algoritmo de Relajación para Estereoscopía

1. **Inicializar**: asignar un uno a todos las correspondencias posibles dentro de un rango de distancias.

2. **Actualizar**: modificar los valores de la matriz de forma que se decrementen elementos que correspondan a la misma línea de vista en el espacio (regla 2) y se incrementan elementos que correspondan a distancias similares (regla 3):

$$C_{t+1}(x,y,d) = k_1[\sum C_t(x',y',d') - k_2 \sum C_t(x'',y'',d'') + C_0(x,y,d)] \tag{6.6}$$

   Donde x', y', d' corresponden a elementos cercanos de la misma disparidad (región excitatoria), x'', y'', d'' a elementos en las mismas coordendadas x, y pero diferente disparidad (región inhibitoria), k_1 y k_2 son constantes. El último término $C_0(x, y, d)$ es el valor inicial de la matriz que contiene todos los posibles apareamientos (no es necesario este término pero si se usa el método converge más rápido).

   Los valores mayores a un límite T se hacen 1 y el resto 0.

3. **Terminar**: repetir (2) hasta que el número de modificaciones sea menor a un número predeterminado N.

El algoritmo funciona muy bien para imágenes de puntos aleatorios como se ilustra en la figura 6.7.

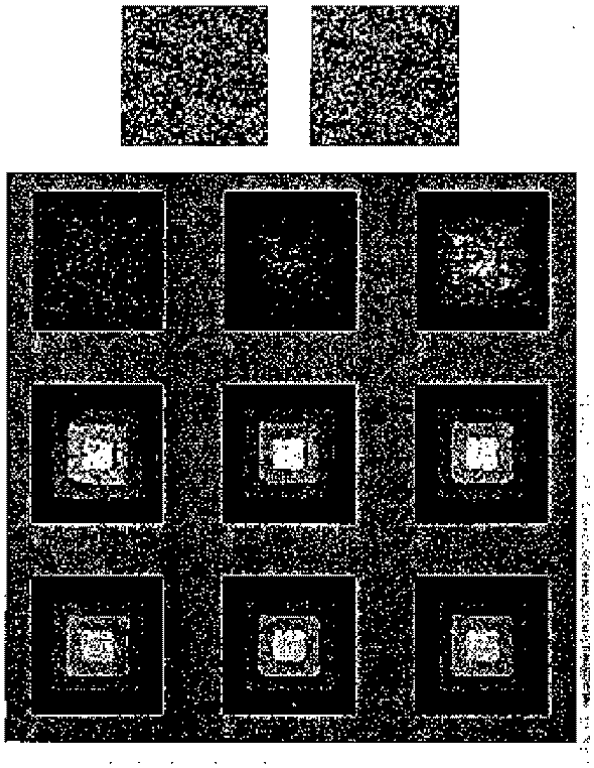


Figura 6.7: Algoritmo de relajación aplicado a un estereograma. En la parte superior se muestran el par estéreo original. Después se ilustra el proceso mediante imágenes de la matriz de disparidad, utilizando diferentes niveles de gris para diferentes disparidades. Se observa que el proceso va desde un estado aleatorio incial, hasta que converge donde se notan claramente 4 regiones concéntricas de diferente nivel de gris, que corresponden a 4 disparidades.

6.3 Forma de sombreado

El mundo esta constituido, en gran parte, por objetos opacos relativamente continuos. Si consideramos una iluminación constante y una reflectividad del objeto aproximadamente uniforme, es posible obtener información de profundidad (3-D) a partir de los cambios de intensidad o sombreado del objeto. A esto se le conoce como "forma de sombreado" (*shape from shading*). Aún considerando dichas simplificaciones el problema es complejo ya que en general existen múltiples soluciones. Para lograr una solución única se han propuesto diversos algoritmos que hacen diferentes consideraciones. Los diferentes algoritmos propuestos los podemos englobar en 3 tipos:

- Uso de múltiples fuentes de iluminación (estereo fotométrico)
- Uso de restricciones entre elementos (relajación)
- Uso de información local (algoritmo diferencial)

En todos los casos consideramos una fuente de iluminación puntual $L(T,U,V)$, un punto en la superficie del objeto (X,Y,Z), y una cámara en (0,0,0) de forma que la imagen coincida con el plano (x,y). Consideramos un sistema de coordenadas (X,Y,Z) centrado en el lente de la cámara de forma que el plano de la imagen (x,y) es paralelo a los ejes (X,Y), y la cámara apunta en la dirección Z (profundidad), como se ilustra en la figura 6.8. En este sistema coordenado, θ es el ángulo entre la luz incidente y la normal a la superficie, ϕ es el ángulo entre la normal y la cámara, y ψ es el ángulo entre la luz incidente y la cámara. En términos vectoriales, $S(T,U,V)$ es la posición de la fuente, $P(X,Y,Z)$ es la posición del punto en el objeto, r es la distancia entre S y P, y $n=[p,q,-1]$ es un vector normal a la superficie.

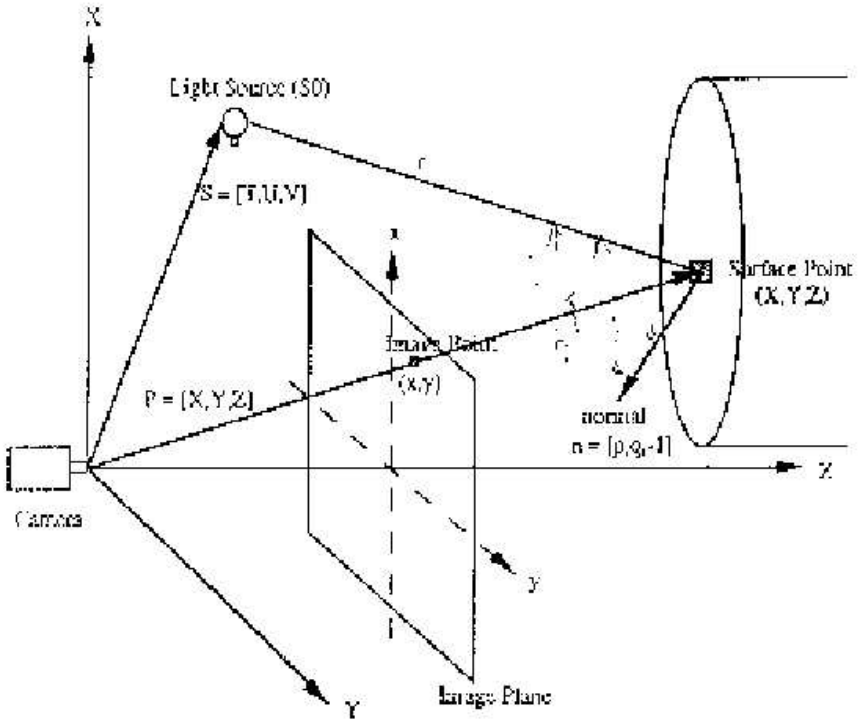


Figura 6.8: Sistema de coordenadas. Se considera la cámara en el origen del sistema de coordenadas del mundo, X,Y,Z, apuntando en el eje Z. El plano de la imagen, x,y, está a una distancia f de la cámara y paralelo al plano X,Y. La fuente de iluminación tiene coordenadas T,U,V, a una distancia r de la superficie. Se considera un punto en el mundo con coordenadas X,Y,Z y vector normal $n=[p,q,-1]$.

El gradiente de la superficie corresponde a su orientación local (relativa). En este sistema de coordenadas, se pueden definir las derivadas parciales de la función de la superficie (Z) respecto a los ejes X, Y:

$$p = \partial Z/\partial x \tag{6.7}$$

$$q = \partial Z/\partial y \tag{6.8}$$

Que corresponden a las componentes en x, y del vector normal.

Una superficie totalmente opaca (*Lambertian*) refleja la luz en la misma magnitud en todas direcciones. Entonces la luz reflejada y, por lo tanto, la intensidad en la imagen, sólo depende de la intensidad de la fuente, S_0, el coeficiente de reflectividad, $r(x, y)$, y el ángulo entre la dirección de la fuente y la normal a la superficie, θ, considerando una fuente lejana:

$$E(x, y) = S_0 r(x, y) cos\theta \tag{6.9}$$

En general se consideran una fuente y coeficientes de reflexión constantes, por lo que:

$$E(x, y) = r_0 cos\theta \tag{6.10}$$

En forma vectorial:

$$cos\theta = n \bullet s/ \mid n \mid \tag{6.11}$$

Donde $s[t, u, v]$ es un vector unitario en dirección de la fuente. Entonces:

$$E(x, y) = r_0 \frac{pt + qu - v}{\sqrt{p^2 + q^2 + 1}} \tag{6.12}$$

Entonces se tiene una ecuación y tres incógnitas (p, q, r_0), por lo que el problema es indeterminado y necesitamos de información adicional para resolverlo. A continuación se describen los 3 principales enfoques utilizados para su solución.

6.3.1 Estereo fotométrico

Un alternativa para obtener mayor información es utilizar múltiples fuentes de iluminación. Cada fuente nos da un valor de intensidad distinto, por lo que si desconocemos la constante r_0 requerimos de 3 fuentes para obtener 3 ecuaciones con 3 incógnitas.

Si denotamos los vectores unitarios de dirección de la fuente k como s_k, tenemos:

$$E_k(x, y) = r_0 n \bullet s_k/ \mid n \mid, k = 1, 2, 3 \tag{6.13}$$

En forma matricial:

$$I = r_0 S n \tag{6.14}$$

Y entonces:

$$n = (1/r_0)S^{-1}I \tag{6.15}$$

La condición para que la matriz sea invertible es que las fuentes no estén en el mismo plano. Otro limitación para que funcione es que no existan sombras para ninguna de las fuentes.

6.3.2 Relajación

El utilizar información local es otra forma de obtener otras restricciones para resolver el problema de forma de sombreado. Para ello se aplica la heurística de que la superficie es suave, es decir, no existen cambios fuertes de profundidad en cierta región.

Una forma de plantear el problema es como un problema de optimización. Para ello se consideran dos aspectos: la información de la ecuación de la intensidad y la información de cambio de intensidad (derivada). Entonces el algoritmo se basa en minimizar una ecuación que toma en cuenta estos dos aspectos. La ecuación a minimizar es la siguiente:

$$e(x,y) = [I(x,y) - E(p,q)]^2 + \lambda[(dp/dx)^2 + (dp/dy)^2 + (dq/dx)^2 + (dq/dy)^2] \tag{6.16}$$

Donde $e(x,y)$ es el término de error a minimizar, $I(x,y)$ es la intensidad del punto y λ es una constante que determina el peso relativo entre la aproximación a la intensidad (primer término) y la "suavidad" de la superficie (segundo término). Derivando la ecuación anterior respecto a p y q se obtiene un sistema de ecuaciones que se puede resolver por métodos numéricos.

6.3.3 Métodos locales

Los métodos anteriores consideran una fuente de iluminación lejana. Si consideramos una fuente cercana, hay que tomar en cuenta la distancia a ésta, por lo que la ecuación de intensidad se convierte en:

$$E(x,y) = S_0 r(x,y) cos\theta / r^2 \tag{6.17}$$

Ya que la intensidad es inversamente proporcional al cuadrado de la distancia (r) entre el objeto y la fuente de iluminación. En forma vectorial, $R = S - P$, $r = \mid S - P \mid$ y $cos\theta = R \bullet n / \mid R \mid\mid n \mid$, por lo que:

$$E(x,y) = S_0 r(x,y)(S-P) \bullet n / \mid n \mid\mid S - P \mid^3 \tag{6.18}$$

Expandiendo los vectores, donde $S = (T,U,V)$, se obtiene:

$$E(x,y) = S_0 r(x,y)[(T-X)p + (U-Y)q - (V-Z)]/(p^2+q^2+1)^{1/2}[(T-X)^2 + (U-Y)^2 + (V-Z)^2]^{3/2} \tag{6.19}$$

El problema se simplifica si consideramos que la posición de la camára es conocida. Sin pérdida de generalidad, podemos situarla en el origen (0, 0, 0) y entonces la ecuación de intensidad se simplifica en:

$$E(x,y) = S_0 r(x,y)[-Xp - Yq + Z]/(p^2 + q^2 + 1)^{1/2}[X^2 + Y^2 + Z^2]^{3/2} \tag{6.20}$$

Para el caso de proyección perspectiva se substituye X por xZ/f, Y por yZ/f, donde f es la longitud focal de la cámara.

$$E(x,y) = S_0 r(x,y)[1 - xp/f - yq/f]/Z^2(p^2+q^2+1)^{1/2}[1+(x/f)^2+(y/f)^2]^{3/2} \tag{6.21}$$

Sin embargo, se pueden tener normalizadas las distancias por la longitud focal ($f = 1$) y se simplifica a $X = xZ$, $Y = yZ$. Substituyendo en la ecuación anterior anterior y factorizando Z:

$$E(x,y) = S_0 r(x,y)[1 - xp - yq]/Z^2(p^2+q^2+1)^{1/2}[1+x^2+y^2]^{3/2} \tag{6.22}$$

Si consideramos que la superficie es suave, podemos considerar que una muy pequeña región es prácticamente plana. Aproximándola por su expansión en series de Taylor al primer grado se obtiene:

$$Z = Z_0 + \partial Z/\partial X(X - X_0) + \partial Z/\partial Y(Y - Y_0) + T.O.S., \tag{6.23}$$

alrededor de un punto X_0, Y_0, Z_0. Por la definición de p y q y despreciando los términos de orden superior (T.O.S.), entonces:

$$Z = Z_0 + p_0(X - X_0) + q_0(Y - Y_0), \tag{6.24}$$

Lo que es igual a:

$$Z = [Z_0 - p_0 X_0 - q_0 Y_0] + p_0 X + q_0 Y \tag{6.25}$$

En términos de coordenadas de la imagen, utilizando proyección perspectiva ($X = xZ$, $Y = yZ$):

$$Z = [Z_0 - p_0 x_0 Z_0/f - q_0 y_0 Z_0/f] + p_0 xZ/f + q_0 yZ/f \tag{6.26}$$

Despejando Z:

$$Z = Z_0 \frac{1 - p_0 x_0/f - q_0 y_0/f}{1 - p_0 x/f + q_0 y/f} \tag{6.27}$$

Substituyendo Z en la ecuación de irradiación (6.20) se obtiene:

$$E(x,y) = \frac{S_0 r(x,y)[1 - xp_0 - yq_0]^3}{Z_0^2(p_0^2+q_0^2+1)^{1/2}(1 - p_0 x_0 - q_0 y_0)^2[1+x^2+y^2]^{3/2}} \tag{6.28}$$

Podemos obtener información adicional considerando el cambio de intensidad de la imagen respecto a x, y: Ex, Ey, los que se pueden obtener mediante operadores diferenciales (Sobel, por ejemplo). De aquí obtenemos dos ecuaciones adicionales que nos permiten resolver el problema.

Utilizando la última ecuación de irradiación (6.28) y considerando las derivadas de intensidad normalizadas (dividiendo entre E), obtenemos:

$$Rx = Ex/E = -3[p_0/(1 - p_0 x - q_0 y) + x/(1 + x^2 + y^2)] \tag{6.29}$$

$$Ry = Ey/E = -3[q_0/(1 - p_0x - q_0y) + y/(1 + x^2 + y^2)] \tag{6.30}$$

De donde obtenemos dos ecuaciones con dos incógnitas que se pueden resolver directamente. $Ex = dE/dx$ y $Ey = dE/dy$ se obtiene como los cambios de intensidad locales mediante algun operador de diferenciación, E es la intensidad promedio en una región "pequeña" y x y y son las coordenadas del punto en le imagen. A partir de estas ecuaciones se obtiene p_0 y q_0 que corresponden al gradiente local de la región correspondiente. Además las constantes (fuente, reflectividad) se han cancelado.

En la figura 6.9 se muestra un ejemplo de la aplicación del método de sombreado local a una imagen. El gradiente resultante se ilustra como pequeños vectores sobre la imagen, los cuales indican la forma relativa del objeto (imagen de gradiente).

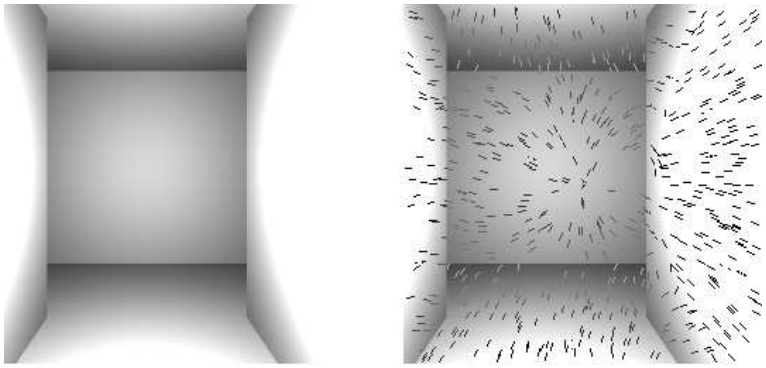


Figura 6.9: Ejemplo de aplicación del método de forma de sombreado local: (a) Imagen sintética de un pasillo. (b) Imagen de los vectores de gradiente.

6.4 Forma de Textura

La textura provee otra alternativa para determinar el gradiente o profundidad relativa de una superficie. Existen 3 alternativas para obtener el gradiente a partir de textura:

1. Razón de cambio máximo de las primitivas de textura.
2. Forma del elemento de textura (texel).
3. Puntos de desvanecimiento

A continuación se describen brevemente cada una de estas 3 técnicas.

Considerando que la textura se puede descomponer en "primitivas", se puede estimar la razón de cambio del tamaño de dichas primitivas en diferentes direcciones. La direccón de la razón de *máximo* cambio corresponde a la dirección del gradiente de textura; la magnitud del gradiente da una indicación de que tanto esta inclinado el plano respecto al eje de la cámara. Este método se ilustra gráficamente en la figura 6.10-a.

Si los *texels* tiene una forma conocida –por ejemplo, círculos–, se puede estimar la orientación de la superficie respecto a la cámara por la deformación de los texels. Por ejemplo, en el caso de *texels* en forma de círculos, la razón entre el eje mayor y menor del elipse resultante de la deformación del círculo, da una indicación de la orientación de la superficie, ver figura 6.10-b.

Cuando se tiene una estructura regular (hileras de *texels*), se puede estimar la orientación a partir de los puntos de desvanecimiento (*vanishing points*). Estos puntos son la proyección al plano

de la imagen de puntos en el infinito a cierta dirección. Considerando 2 puntos de desvanecimiento –textura rectangular, la línea que une ambos puntos provee la orientación de la superficie, como se ilustra en la figura 6.10-c.

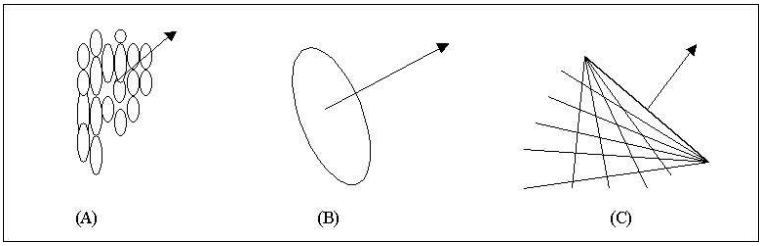


Figura 6.10: Técnicas para la obtención de forma a partir de textura: (a) orientación por gradiente de textura, (b) orientación por deformación de texels, (c) orientación por puntos de desvanecimiento.

6.5 Referencias

El trabajo en obtención de profundidad o forma de "X" tiene una larga tradición en visión computacional. Existen algunos textos dedicados específicamente a este tema, de los recientes destacan el de Trucco y Veri [119] y el de Faugeras [21].

Entre los trabajos inciales importantes, destaca el de [77], quien propone el método de relajación para estereo. También es importante el incio de la línea de forma de sombreado por Horn [38, 39] (ver también [40] para una revision histórica). La técnica de sombreado local descrita en este capítulo se basa en el desarrollo de Rashid y Burger [97].

6.6 Problemas

1. ¿Porqué se dice que se pierde la tercera dimensión en una imagen? ¿Qué alternativas hay para recuperar la dimensión perdida?

2. Dibuja mediante pequeños vectores el gradiente (p,q) para una esfera, un cilindro y un cubo. A esto se le conoce como "diagramas pq".

3. ¿De qué factores depende la intensidad de un punto (pixel) en la imagen? ¿Cómo se puede utilizar esta información para recuperar la profundidad (3-D)?

4. Realiza el desarrollo matemático completo para obtener las ecuaciones para Rx y Ry, en el método local, a partir de la ecuación de intensidad.

5. Dada la ecuación de intensidad para el método de forma de sombreado: ¿Qué consideraciones se hacen y qué información adicional se utliza para resolverlo en forma directa? ¿Qué se obtiene con éste método?

6. Considerando el caso de una fuente lumínica en el mismo punto de la cámara y superficies mate. Obten el valor del gradiente (p, q) para: (a) una imágen de intensidad constante, (b) una imágen cuya intensidad aumenta linealmente en dirección x. Demuestra que concuerda con los valores obtenidos por el método local visto en clase.

7. Considera la siguiente imagen estereo de puntos aleatorios. Obten la disparidad mediante el método de relajación. Muestra el desarrollo.

1	0	0	0		1	0	0	0
0	1	1	0		0	0	1	1
0	1	1	0		0	0	1	1
0	0	0	0		0	0	0	0

8. Considera la siguiente imagen estereo de puntos aleatorios: (a) Obten la matriz inicial de disparidad para los "1" considerando que las cámaras están alineadas horizontalmente y una máxima disparidad de 2 pixels. (b) Calcula la matriz después de un ciclo del algoritmo de relajación. Indica que valor seleccionaste para los diferentes parámetros necesarios en el método.

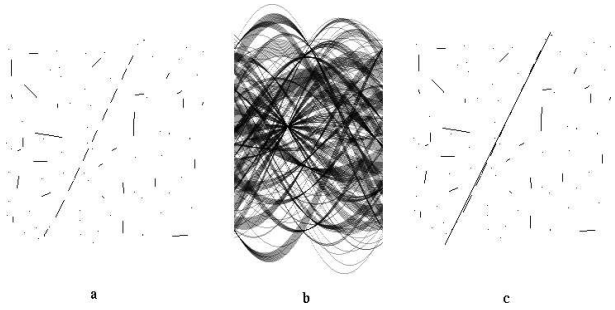
0	0	0	0		1	0	0	0
0	1	1	0		1	1	0	0
0	1	1	0		1	1	0	0
0	1	1	0		1	1	0	0

9. Repite el problema anterior utilizando la técnica de correlación, considerando solamente la correlación en sentido horizontal. Indica el método que seleccionaste para calcular la correlación, y obten la matriz de disparidad.

10. Dada una textura con texels en forma de círculos, plantea como estimar cuantitativamente la orientación de la superficie en base a la deformación de los texels.

6.7 Proyectos

1. Implementar en en el laboratorio visión estereo mediante la técnica de correlación. Para ello:
   - obtener las orillas verticales de las imágenes (pares estereo) mediante algun operador direccional como Sobel,
   - obtener la disparidad mediante el enfoque de correlación considerando una región sobre las imágenes de orillas (restringir a solo correlación horizontal a una distancia máxima),
   - mostrar los resultados mediante una imagen de disparidad,
   - probar con diferentes pares estereo y deplegar la imagen de disparidad obtenida.
2. Implementar en en el laboratorio visión estereo mediante la técnica relajación de Marr, probar con imágenes estero de puntos aleatorios. Seguir un proceso similar al del proyecto anterior, pero si la obtención de orilla (directamente sobre los pixels de la imagen original).



Capítulo 7

Agrupamiento de orillas

7.1 Introducción

En los capítulos anteriores hemos visto como obtener ciertos atributos de las imágenes, como son las orillas, color, textura, profundidad y movimiento; que comprenden los conoce como visión de niviel bajo. En este capítulo y el siguiente abordaremos la visión de nivel intermedio. El propósito de la visión de *nivel intermedio* es generar una representación más compacta de lo que puede ser detectado en la visión de bajo nivel. Tales representaciones deben ser más útiles que trabajar con miles de pixeles. Dos maneras de reducir esta cantidad de información son *agrupar las orillas* para producir bordes y el determinar las regiones en la imagen. La búsqueda de regiones se basa en la suposición de que objetos del mundo tienen características aproximadamente similares a lo largo de su superficie. Estas regiones se manifiestan de dos manera (ver figura 7.1), la *región misma* a tratar y el *borde* que divide una región con las vecinas. Encontrar los bordes es escencial para delimitar regiones y viceversa.

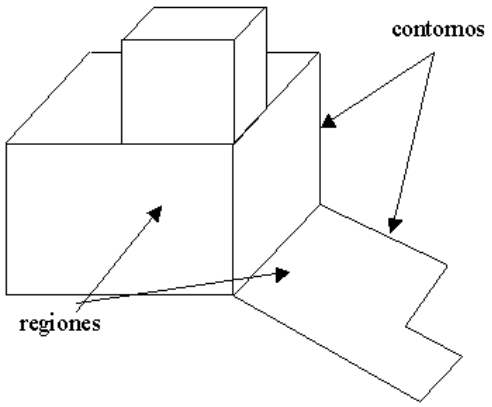


Figura 7.1: Segmentación. Se muestran las dos formas bśicas de segmentar una imagen: mediante los contornos o bordes y mediante las regiones.

En este capítulo veremos como generar bordes a partir de orillas y en el siguiente como determinar las regiones. Aunque existen múltiples técnicas para la generación de bordes, aquí analizaremos tres de las más representativas:

- transformada de Hough,
- búsqueda en grafos,
- agrupamiento perceptual.

Antes analizaremos una forma de estructurar las imágenes en pirámides y árboles cuaternarios (*Quadtrees*) que es útil para varias técnicas.

7.2 Pirámides y árboles cuaternarios (*Quadtrees*)

Una imágen cualquiera la podemos considerar a diferentes niveles de resolución. La mayor resolución la obtenemos al considerarla a nivel *pixels*. Éstos los podemos agrupar obteniendo imágenes de menor resolución hasta llegar a un sólo elemento que represente toda la imagen. De esta forma se obtiene una estrctura piramidal, dónde en la base de la pirámide se tiene los *pixels* y en la cima la imagen total, y entre éstos, la imagen a diferentes resoluciones. La figura 7.2 muestra esta estructura.

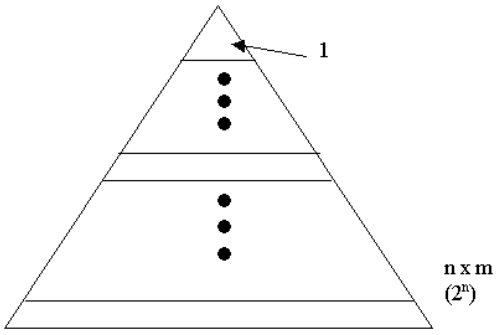


Figura 7.2: Estructura piramidal. En el nivel inferior de la pirámide se tiene la imagen a su máxima resoulción de $n \times m$ (se simplifica si se considera de $2^N \times 2^N$). La resolución de la imagen va disminuyendo al ir aumentando de nivel en la pirámide hasta llegar a un solo valor (resolución mínima) en la punta.

Una forma de obtener dicha estructura piramidal es mediante la división sucesiva de la imagen en cuadrantes. Se divide la imagen en cuatro rectángulos (para facilitar los calculos se utilizan cuadrados) iguales, estos as su vez en cuatro y así hasta llegar al nivel pixel. Cada cuadrante se puede ver formado por cuatro "hijos" correspondientes al nivel inferior, de forma que toda la estructura forma un árbol de grado cuatro –árbol cuaternario o *quadtree*–, como se ilustra en la figura 7.3.

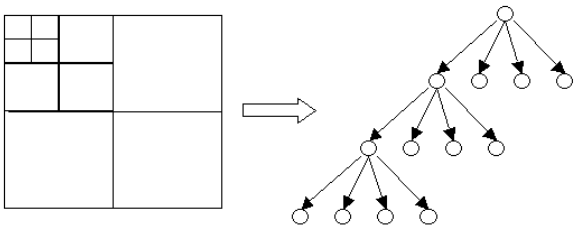


Figura 7.3: Árbol Cuaternario.

Cada elemento de un nivel depende de sus cuatro hijos. Normalmente se obtiene simplemente mediante el promedio de intensidades, lo que equivale a una digitalización de menor resolución, como se ilustra en la figura 7.4. Existen otras formas de combinar las elementos para construir el árbol, como el utilizar una representación binaria o restringir los nivels de gris. La representación se simplifica si consideramos imágenes de dimensión $2^N \times 2^N$.

La figura 7.5 muestra las particiones de un árbol cuaternario aplicado a una imagen sintética. Los cuadros se dividieron hasta una área mínima de 4x4 pixeles. La decisión de particionar esta

Figura 7.4: Ejemplo de una imagen a diferentes niveles: (a) imagen original (máxima resolución); (b), (c), (d) imágenes a diferentes niveles en la pirámide (menor resolución).

dada por las diferencias en la desviación estándar. En la práctica, cuando el árbol cuaternario se construye sobre imágenes reales, ver por ejemplo la figura 7.6, se generan grandes cantidades de particiones, las cuales en muchas ocasiones deben ser juntadas para describir algo más útil. Esta técnica de partición y juntar (split & merge) será vista en el siguiente capítulo.

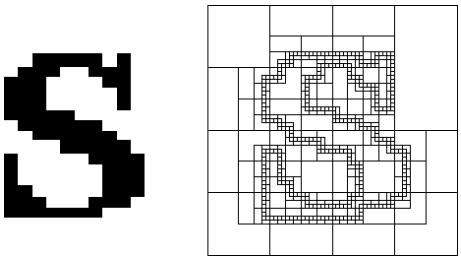


Figura 7.5: Regiones homogéneas: particiones de un árbol cuaternario aplicado a una imagen sintética.

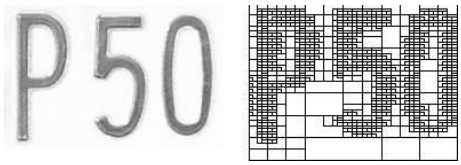


Figura 7.6: Regiones homogéneas: particiones de un árbol cuaternario aplicado a una imagen real.

En ciertas aplicaciones es conveniente que los cuadrantes se traslapen, de forma que exista intersección entre ellos. Si consideramos un traslape del 50%, entonces tenemos que cada nodo intermedio tiene 4 padres y 16 hijos. Éste caso se muestra en la figura 7.7.

La representaciones piramidales basadas en árboles cuaternarios se han utilizado en diversas técnicas a diferentes niveles de visión, como en detección de orillas, segmentación, reconocimiento de forma, etc.

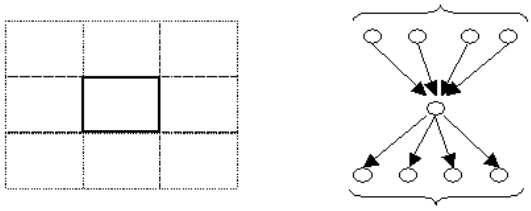


Figura 7.7: Pirámide traslapada.

7.3 Transformada de Hough

La *transformada de Hough* está orientada a la detección de contornos cuya forma básica es conocida y que puede ser representada como una curva paramétrica, tales como líneas, círculos, elipses, cónicas, etc.

Primero consideremos el caso de una línea recta. Se tienen varios puntos (orillas) que tienen una alta probabilidad de pertenecer a una línea, pero existen algunas orillas faltantes y otros puntos fuera de la línea. El objetivo es encontrar la ecuación de la línea que "mejor" explique los puntos existentes, ver figura 7.8.

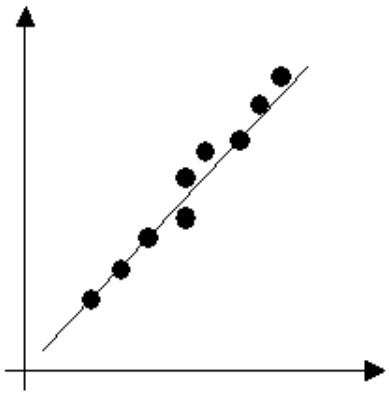


Figura 7.8: Detección de líneas. Se ilustra la recta que mejor aproxima el borde descrito por los puntos que representan orillas.

Para esto consideremos la ecuación de una línea, que es:

$$y = mx + b \tag{7.1}$$

Si consideramos una orilla (x_1, y_1), ésta puede pertenecer a todas las líneas posibles que pasen pasen por dicho punto, es decir todas las (m, b) que satisfagan la ecuación:

$$y_1 = mx_1 + b \tag{7.2}$$

Entonces podemos pensar que hacemos una transformación del espacio $x - y$ al espacio $m - b$, conocido como *espacio paramétrico*. Un sólo punto en el espacio de la imagen corresponde a un número infinito de puntos en el espacio paramétrico (una línea), como se muestra en la figura 7.9.

Considerando dos puntos en dicha línea:

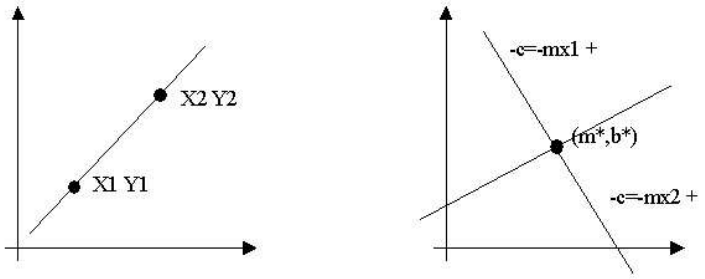


Figura 7.9: Espacio de la imagen y espacio paramético. Cada punto en el espacio de la imagen, x_1, y_1, x_2, y_2 (figura izquierda), corresponde a una línea en el espacio paramétrico (figura derecha). Donde se intersectan dichas líneas, son los parámetros de la recta que los une en el espacio de la imagen.

$$(x_1, y_1), y_1 = mx_1 + b \tag{7.3}$$

$$(x_2, y_2), y_2 = mx_2 + b \tag{7.4}$$

Obtenemos dos líneas en el espacio paramétrico y su intersección nos da los parámetros (m, b) de la línea que buscamos. Sin embargo, en las imágenes reales, normalmente hay puntos (orillas) faltantes o puntos adicionales por ruido, por lo que es necesario considerar todas las orillas presentes en la imagen (o el área de interés) para tener una estimación más robusta.

En la práctica se discretizan los parámetros (m, b) en un número limitado de valores, formando una matriz bidimensional en el espacio paramétrico llamado *acumulador*:

$$A(m, b) \tag{7.5}$$

Dicho acumulador se inicializa a cero. Cada orilla (mayor a un límite o como el máximo en la dirección del gradiente) contribuye a una serie de valores en el acumulador, sumándole una unidad a las combinaciones posibles de (m, b):

$$A(m, b) = A(m, b) + 1, y_i = mx_i + b \tag{7.6}$$

El elemento del acumulador que tenga un número mayor (más votos) corresponde a la ecuación de la línea deseada $(m*, b*)$. En la figura 7.10 se ilustra el acumulador, suponiendo 5 valores para cada parámetro, (m, b), después de que se han considerado varias orillas.

1	1	1	1	0
0	2	1	0	0
0	4	1	0	0
1	1	0	1	0
1	1	0	0	1

Figura 7.10: Ejemplo del acumulador, $A(m, b)$, con 5 particiones por parámetro. Se muestra después de incluir varias orillas, donde el elemento con más votos es el $[3, 2]$.

También se pueden detectar, en principio, N líneas rectas en la imagen considerando los N mayores elementos del acumulador o los que tengan un cierto número de votos mayor a un umbral.

Dado que m puede tomar valores infinitos, normalmente se utiliza una parametrización más conveniente de la línea recta:

$$xcos(\theta) + ysen(\theta) = \rho \tag{7.7}$$

Donde ρ es la distancia al origen de la recta (en forma perpendicular) y θ es el ángulo respecto al eje X. El espacio paramétrico, (θ, ρ), es diferente, pero el método es el mismo. Esta representación es la utilizada en la práctica para implementar la transformada de Hough para detección de rectas.

La figura 7.11 muestra un ejemplo de la transformada de Hough para la detección de rectas. Como puede verse, esta técnica permite agrupar las orillas en bordes, tolerando la ausencia de ciertas orillas y ruido en la imágen.

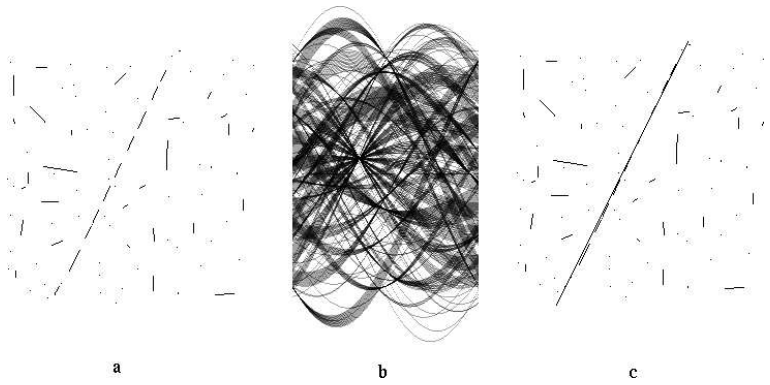


Figura 7.11: Ejemplo de la transformada de Hough. (a) Imagen original. (b) Espacio paramétrico (θ, ρ). (c) Superposición de la mejor recta encontrada ($\theta = 64^o$).

La técnica de la transformada de Hough se puede extender a otro tipo de curvas (círculos, elipses, etc.) simplemente tomando su ecuación y utilizando el espacio paramétrico correspondiente. Aún cuando está orientada a detectar curvas paramétricas, existen algunas extensiones que permiten su aplicación a curvas no paramétricas.

Otra extensión consiste en combinarla con una representación en base a árboles cuaternarios. Para ésto se divide la imágen en una serie de ventanas (que pueden ser a diferentes tamaños y traslapadas) y se aplica la transformada de Hough en cada ventana para detectar secciones de línea. Dichas secciones pueden ser posteriormente integradas en líneas o curvas uniendo los segmentos de cuadrantes continuos. Para ellos se aplica la estructura de árbol, indicando en cada nodo la prescencia de segmentos de línea en nodos inferiores y de esta forma optimizar la integración de los segmentos a diferentes niveles. Un ejemplo de la combinación de la transformada de Hough combinada con *Quadtrees* se ilustra en la figura 7.12.

La transformada de Hough puede ser extendida para utilizar información de la dirección del gradiente, lo que disminuye los votos de cada punto y reduce su complejidad. También se puede usar para siluetas cuya forma es conocida *a priori* pero no son curvas paramétricas (transformada de Hough generalizada).

7.4 Técnicas de búsqueda

Las técnicas de búsqueda para agrupamiento de orillas se basan en considerar una imagen de orillas (magnitud y dirección) como un grafo pesado. Entonces, el encontar un contorno se puede ver como un proceso de búsqueda en grafos.

Consideramos una imágen de orillas, con magnitud s y dirección θ. Cada orilla (mayor a un límite o como el máximo en la dirección del gradiente) la consideramos un nodo de un grafo con peso s, y cada nodo se conecta a otros nodos de acuerdo a la dirección del gradiente. Normalmente se considera conectado a los 3 pixels (nodos) vecinos en la dirección del gradiente. De esta forma se

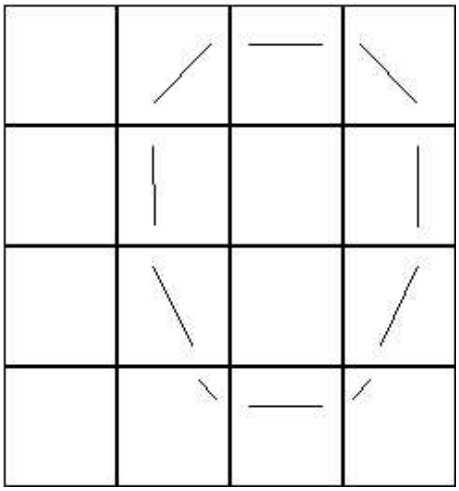


Figura 7.12: Transformada de Hugh combinada con *QuadTrees*.

puede construir un grafo a partir de una imagen de orillas. Se parte de una orilla inicial (arbitraria) que corresponde al nodo inicial del grafo, y este se conecta a otras orillas en base a la dirección del gradiente, constituyendo las aristas del grafo. El grafo así obtenido representa las orillas y su relación de vecindad, como se muestra en la figura 7.13.

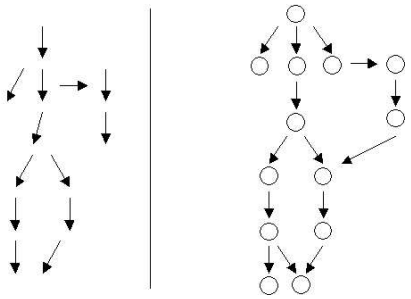


Figura 7.13: Imagen de gradientes y su gráfica correspondiente.

Entonces el encontrar un contorno de un punto inicial e_s a un punto final e_f se puede plantear como la búsqueda de una trayectoria entre dichos nodos en el grafo, la cual debe ser óptima respecto a cierto criterio de evaluación. Entre las funciones de evaluación para encontrar la trayectoria óptima están las siguientes:

- Magnitud de la orillas.
- Curvatura, diferencia entre las direcciones de gradiente de orillas continuas en el grafo.
- Proximidad, distancia a la posición aproximada del borde (en este caso se asume que se tiene un estimado incial de la posición del borde *a priori*).
- Distancia a la meta (se asume conocimiento del punto final del borde).

Para esto se pueden utilizar diferentes técnicas de búsqueda, incluyendo métodos exhaustivos (búsquedas por profundidad y a lo ancho) y búsquedas heurísticas. Para una búsqueda heurística se define una función de costo para cada arco, buscándose la trayectoria de menor costo. Esta se basa en las funciones de evaluación anteriores. Para evitar problemas de orillas faltantes se pueden interpolar elementos antes de la búsqueda, o modificar la definición de vecinos para permitir saltar pixels. La técnica puede extenderse para encontar todos los contornos (todas las trayectorias posibles).

7.5 Agrupamiento perceptual

Las técnicas de *agrupamiento perceptual* se basan en teorías sobre la forma en que los humanos manejamos características para segmentar o segregar objetos en escenas. Algunos de sus origenes se encuentra en la escuela Gestalt de psicología.

Existen una serie de principios o reglas heurísticas en las cuales se supone se basa nuestra percepción para agrupar elementos en contornos o regiones. Algunas de las reglas o principio para agrupamiento perceptual son las siguientes:

- *Proximidad*, elementos cercanos tienden a ser percibidos como una unidad.
- *Similaridad*, elementos similares (en intensidad, color, forma, etc.) tienden a ser parte de una unidad.
- *Continuidad*, elementos forman grupos que minimizan el cambio o discontinuidad.
- *Cerradura*, elementos se agrupan en figuras completas regulares.
- *Simetría*, regiones rodeadas por contornos simétricos se perciben como figuras coherentes.
- *Simplicidad*, si existe ambigüedad, de forma que se pueden percibir dos o más figuras de los mismos elementos, ésta se resuelve en favor de la alternativa más simple.

En la figura 7.14 se ilustran en forma gráfica algunos de los principios de organización perceptual.

Estos principios se pueden aplicar para agrupar orillas en contornos, incluso en la prescencia de ruido, oclusión u otros fenómenos. Para agrupamiento de orillas, comunmente se aplican tres principios:

1. Proximidad, orillas cercanas y/o que forman segmentos de línea recta.
2. Continuidad, orillas que forman líneas o curvas continuas.
3. Similaridad, orillas similares en intensidad, contraste u orientación.

Estos principios pueden ser combinados para agrupar orillas, resultando en una técnica bastante robusta que puede ser aplicada a cualquier clase de contornos. También es posible combinar estructuras piramidales para extraer contornos, en este caso se puede aplicar en dos niveles:

1. Para la extracción de segmentos de línea en las ventanas, eliminándo aquellos que no satisfagan los principios
2. Para conectar los segmentos de línea entre ventanas y formar contornos. Para esto se utilizan los criterios de continuidad y similaridad en orientación, conectando aquellos segmentos continuos (distancia máxima) cuya orientación sea similar dentro de cierto rango (por ejemplo 45 grados).

El agrupamiento perceptual en una estructura piramidal se ilustra en la figura 7.15 con dos ejemplos, uno aplicado a imágenes sintéticas con ruido y otro a imágenes reales obtenidas de un endoscopio.

El agrupamiento perceptual se puede considerar como una especie de post–procesamiento a las orillas que se obtienen con un detector de orillas, manteniendo las orillas que satisfagan las reglas anteriores.

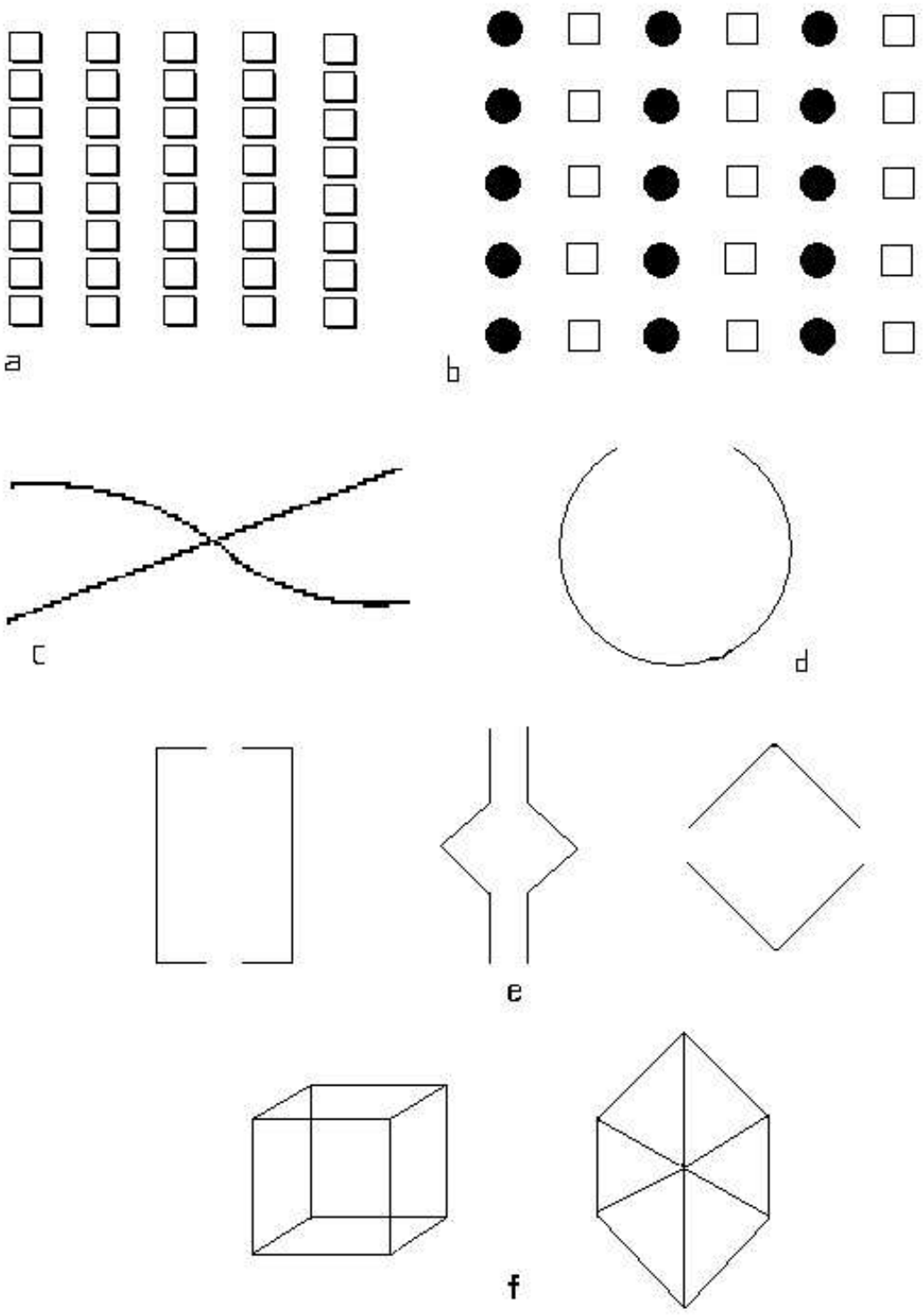


Figura 7.14: Algunos principios de la organización perceptual: (a)*Proximidad*, (b)*Similaridad*, (c)*Continuidad*, (d)*Cerrado*, (e)*Simetría*, (f)*Simplicidad*.

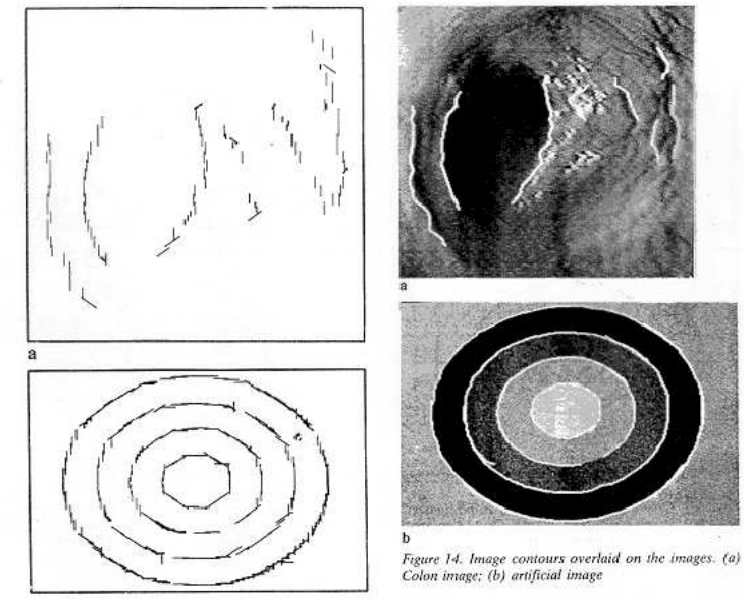


Figura 7.15: Ejemplos de agrupamiento perceptual. En la parte superior se ilustra un ejemplo de una imagen del interior del colon obtenida con un endoscopio, en la parte inferior una imagen sintética con ruido. En ambos casos se muestran las orillas obtenidas (izquierda) con el método de agrupamiento perceptual, y las orillas sobrepuestas en blanco (derecha) sobre la imagen original.

7.6 Referencias

El problema de segmentación ha sido reconocido desde hace tiempo como uno de los problemas fundamentales y más difíciles en visión por computadora. En particular, en el enfoque de segmentación mediante encontrar los bordes que separan a las regiones o agrupamiento de orillas, se ha realizado investigación desde los 60's. Ballard [2] incluye un capítulo dedicado a la detección de bordes.

La transformada de Hough fue propuesta originalmente, por supuesto, por Hough [41]. La introducción de la transformada de Hough a visión por computadora se debe a R. Duda y P. Hart [18]. Posteriormente, D. Ballard [3] desarrolló la extensión para curvas en general o lo que se conoce como la transformada de Hough generalizada. Las transformada de Hough continua siendo tema de investigación [138]. Algunos de los temas de interés son el reducir el costo computacional [52], diseñar implementaciones paralelas [68], utilizar modelos no deterministicos [78, 60, 49] o investigar que otras propiedades se pueden detectar con aplicacione sucesivas sobre el espacio paramétrico [125].

Lester y otros [**?**] desarrollaron la técnica de búsqueda basada en grafos, y la aplicaron al problema de detectar células blancas en imágenes.

Dos de las referencias base para el agrupamiento perceptual, basados en la escuela Gestalt, son los libros de D. Lowe [71] y G. Kanizsa [115]. Uno de los trabajos mas famosos en la segmentación de objetos en base a explotar el agrupamiento perceptual es el presentado por A. Shashua y S. Ullman [104]. En el se calcula una función que permite extraer el objetos más largos y con cambios suaves. Las segmentaciones utilizando esta "saliencia estructural" son particularmente interesantes, pero no ha resultado fácil [1] extender este modelo a más de un objeto por escena. Khan y Gillies [58] utilizarón el enfoque de agrupamiento perceptual, en combinación con una representación piramidal, para la detección de curvas.

La mayoría de estas pistas perceptuales no se han implementado realmente en visión ya que resulta complicado completar los objetos ocluídos o inconclusos en imágenes reales con el principio de cerradura. Sin embargo, algunas publicaciones han mostrado, en imágenes sintéticas, que es posible completar los objetos utilizando el principio de cerradura [34]. Actualmente el modelo de integración de características mas aceptado es el de Wolfe [137] en donde se toman en cuenta relaciones de alto y bajo nivel (*top-down y bottom-up*). Computacionalmente hablando, una manera de utilizar estas características es sumando votos en un acumulador bidimensional, llamado mapa saliente [63, 44, 130, 43]. Los lugares con más votos o más *salientes* son posteriormente analizados para extraer objetos de interés.

7.7 Problemas

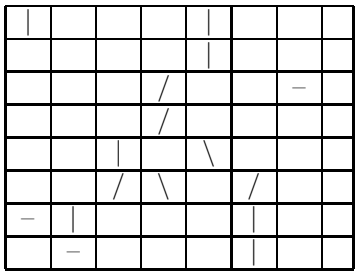
1. ¿Qué desventajas tiene la transforma de Hough en su forma original? ¿Cómo se pueden reducir dichos problemas? ¿En qué consiste la transformada de Hough *generalizada*?

2. Dada la siguiente imagen:

1	1	1	0
0	1	1	1
0	0	1	1
0	0	0	1

   (a) Obten la magnitud de las orillas aplicando el operador de Sobel, muestra la imagen resultante. (b) Considerando sólo las orillas mayores a un umbral (especificar), obten en el espacio (ρ, θ) la matriz acumulador aplicando la transformada de Hough. (c) Indicar los parámatros de la línea más notoria de la imagen.

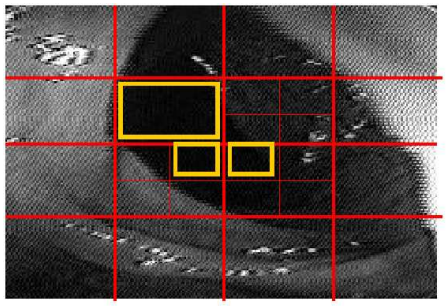
3. Considera la siguiente imagen de orillas:



Obten la "línea" que las agrupe mediante la transformada de Hough. Muestra el desarrollo.

4. Para la imagen de orillas del problema anterior, aplica la técnica de búsqueda en grafos para obtener el contorno. Define una heurística y con ella selecciona la "mejor" trayectoria en el grafo.

5. A imágenes de dibujos planos con líneas rectas continuas y punteadas, se les ha aplicado detección de orillas tipo Sobel y se tiene la magnitud y dirección de cada punto. Considerando que se han adelgazado las orillas y no hay ruido, describe un algoritmo para agrupar las orillas y formar líneas de forma que éstas se describan como "línea de tipo (continua o punteada) de X_1 a X_2".



Capítulo 8

Segmentación

8.1 Introducción

El separar la imagen en unidades significativas es un paso importante en visión computacional para llegar al reconocimiento de objetos. Este proceso se conoce como *segmentación*. Una forma de segmentar la imagen es mediante la determinación de los bordes, que se analizó en el capítulo anterior. El dual de este problema, es determinar las regiones; es decir, las partes o segmentos que se puedan considerar como unidades significativas. Esto ayuda a obtener una versión más compacta de la información de bajo nivel, ya que en vez de miles o millones de pixels, se puede llegar a decenas de regiones. Las caraterísticas más comúnes para delimitar o *segmentar* regiones son: intensidad de los pixeles, textura, color, gradiente y profundidad relativa.

Una suposición importante, que normalmente se asume en visión de nivel intermedio, es considerar que pixeles de un mismo objeto comparten propiedades similares. Por ejemplo, al procesar una imagen de una manzana, suponemos que el color de sus pixeles es aproximadamente homogéneo. En la vida real esto no es totalmente cierto, el color de los pixeles varía. Para evitar este tipo de variaciones es mejor considerar un color "aproximadamente" similar sobre una *región* mas que a nivel pixel. Encontrar este tipo de regiones homogéneas es el tema principal de este capítulo. Este no es un problema sencillo, ya que es difícil distinguir las variaciones propias del objeto o por cambios de iluminación (por ej., sombras), de las diferencias por tratarse de otro objeto. En la figura 8.1 se muestra una imagen sencilla, en la que se distinguen claramente las diferentes regiones.

Figura 8.1: Ejemplo de imagen con las regiones significativas. Cada huevo de diferente color corresponde a una región, además se tiene otras 2 regiones que corresponden al canasto y el fondo

Existen varias técnicas para la segmentación de regiones. Éstas se pueden clasificar en tres tipos:

1. Locales – se basan en agrupar pixels en base a sus atributos y los de sus vecinos (agrupamiento).
2. Globales – se basan en propiedades globales de la imagen (división).
3. División–agrupamiento (*split & merge*) – combinan propiedades de las técnicas locales y globales.

En este capítulo, las técnicas serán analizadas considerando, principalmente, el nivel de intensidad de pixels como la característica para delimitar las regiones. Sin embargo,, como se mencionó anterioremnte, pueden utilizarse otros atributos.

8.2 Segmentación por histograma

La segmentación por histograma (*thresholding*) es una técnica global que se basa, inicialmente, en asumir que hay un sólo objeto sobre un fondo uniforme. Por esto se consideran dos regiones en la imagen y para dividirlas se toma como base el histograma de intensidades.

Podemos asumir que si hay dos regiones se tiene dos picos en el histograma. Entonces se toma el valle (mínimo) entre los dos y este se considera la división entre las dos regiones. De esta forma todos los pixels que correspondan a un lado del histograma se toman como una región y el resto como otra, como se ilustra en la figura 8.2.

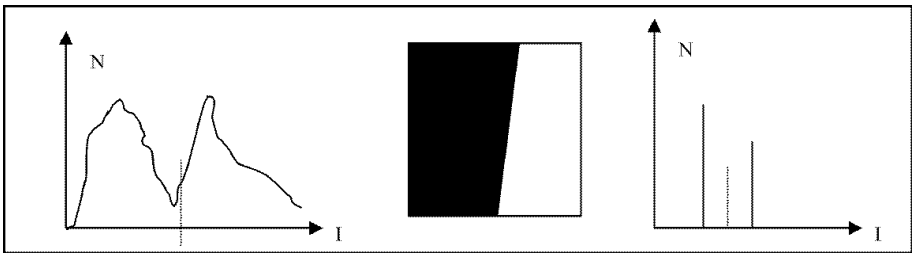


Figura 8.2: Segmentación por histograma. De lado izquierdo se muestra un histograma típico de una imagen con dos regiones (bimodal). La imagen del centro es un ejemplo de una imagen "ideal" con dos regiones, la cual produce el histograma del lado derecho. En ambos histogramas se indica con una línea punteada la separación del histograma, que corresponde a las dos regiones de la imagen.

Cabe notar que esta técnica sólo considera la intensidad (u otro atributo) de los pixels, sin tomar en cuenta la coherencia espacial de la región. Por ello, dos pixels separados en la imagen, pueden pertenecer a la misma región si tienen intensidades similares. Un ejemplo de segmentación por histograma se muestra en la figura 8.3, con diferentes puntos de división (umbrales). Este ejemplo ilustra la importancia de la determinación del punto de división (mínimo) en el histograma.

La técnica de segmentación por histograma se puede extender a N regiones, tomando cada pico del histograma como una región y los mínimos correspondientes como las divisiones entre regiones. Esto se ilustra en la figura 8.4. En la práctica, esta forma de segmentación funciona para "pocas" regiones, ya que al aumentar éstas, se vuelve muy difícil determinar los picos y valles del histograma.

Otra variación de este algoritmo es su aplicación a imágenes de color. Para ello se obtiene el histograma para cada componente de color en diferentes modelos, se hace la división en cada histograma y se combinan los resultados. Esto se realiza mediante una división recursiva de la siguiente manera:

1. Considerar cada región (inicialmente toda la imagen) y obtener los histogramas para cada componente.

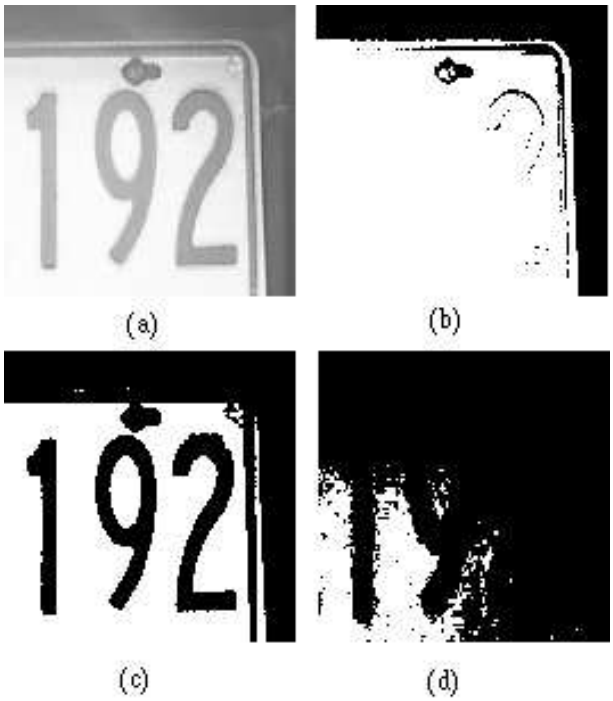


Figura 8.3: Ejemplo de segmentación por histograma. (a) Imagen original. (b) Segmentación con división en 150. (c) Segmentación con división en 200. (d) Segmentación con división en 250.

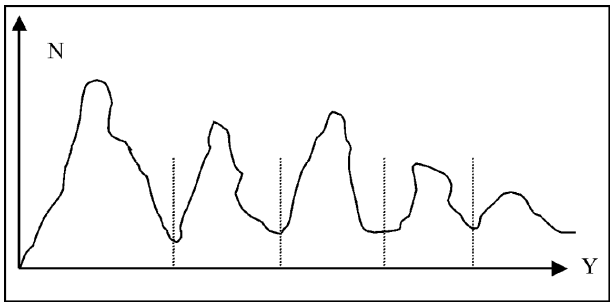


Figura 8.4: Histograma de una imagen con múltiples regiones. Cada valle, indicado con una línea punteada, representa la división entre dos regiones.

2. Tomar el pico más significativo de las componentes y utilizarlo para dividir la región en 3 subregiones tomando los dos mínimos en cada lado del pico.
3. Repetir los pasos (1) y (2) hasta que ya no existan picos significativos.

Un ejemplo de aplicación de ésta técnica a una imagen de un paisaje a color se muestra en la figura 8.5. Éste algoritmo ha sido aplicado para la segmentación de imágenes aéreas de satélite (LANDSAT).

Aunque se puede aplicar en imágenes simples, esta técnica presenta ciertas limitaciones:

- Es difícil identificar correctamente los mínimos en el histograma.
- Se tienen problemas cuando las regiones varíán suavemente su nivel (sombreado, por ejemplo).
- Se aplica sólo cuando hay pocas regiones.
- No se pueden distinguir regiones separadas de niveles similares de gris (conectividad).

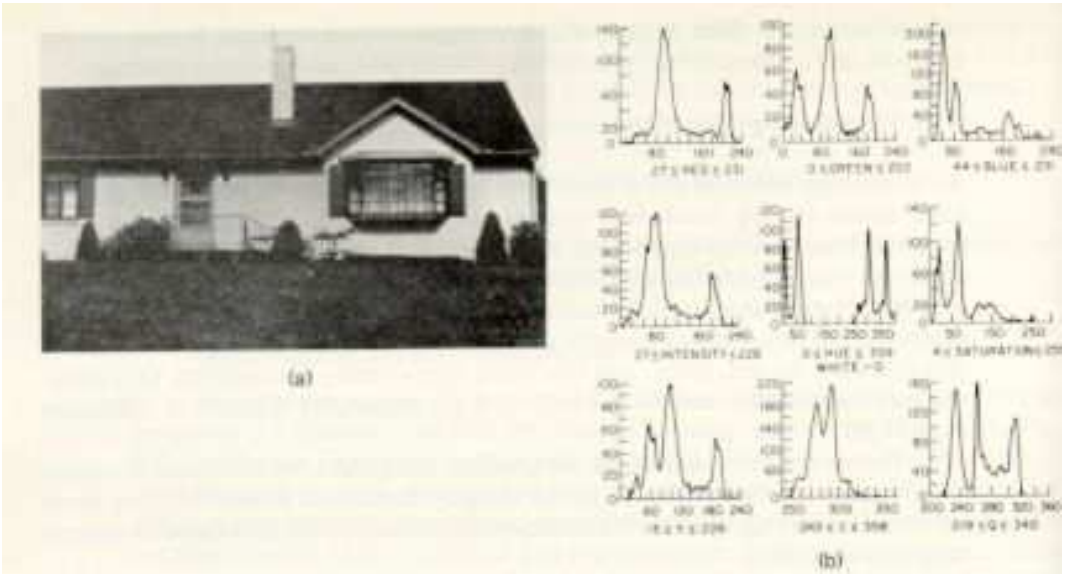


Figura 8.5: Segmentación por histograma de imágenes a color. Del lado izquierdo se muestra la imagen original, y del lado derecho los histogramas de la imagen en 3 diferentes modelos de color.

Algunas de estas limitaciones se ilustran en la figura 8.6. A continuación analizaremos otras técnicas que atacan algunos de estos problemas.

8.3 Segmentación por crecimiento de regiones

Los métodos de segmentación por *crecimiento de regiones*, son técnicas locales que se basan en tomar un *pixel*, o conjunto de *pixels*, como una región inicial (semilla) y a partir de éstos "crecer" la región con puntos similares hasta llegar a ciertos límites, como se ilustra en la figura 8.7.

Para el crecimiento de regiones existen dos problemas básicos:

1. Selección de los puntos iniciales. Para esto se puede tomar ciertos puntos específicos de acuerdo a información previa (más negro, más brillante), o buscar grupos de puntos muy similares y tomar su centroide como punto inicial.

2. Criterio de similitud. Para esto se toma alguna heurística de diferencia máxima entre pixels, junto con criterios de conectividad. Se pueden usar otros criterios como el número de regiones o su dimensión esperada, o información del histograma.

Un ejemplo sencillo de crecimiento de regiones se muestra en la figura 8.8.

Dentro de las técnicas de segmentación local existen diversas variantes dependiendo de la representación de la imagen. Consideraremos dos técnicas, una basada en una representación de estados y otra basada en grafos.

8.3.1 Método de búsqueda en espacio de estados

Bajo el enfoque de una representación de espacio de estados, se considera a la imagen como un "estado discreto", donde cada *pixel* es una región distinta. Se cambia de estado al insertar o remover una división entre regiones. Entonces el problema se convierte en una búsqueda en el

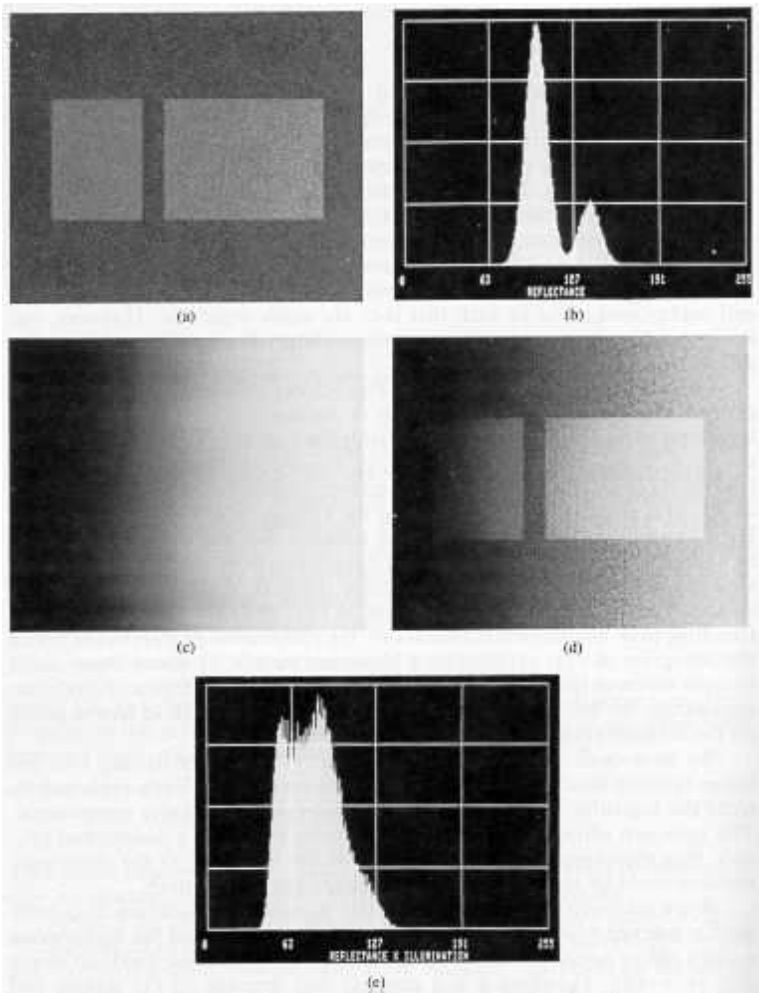


Figura 8.6: Ejemplos de las limitaciones de segmentación por histograma. En la parte superior, se ilustra una imagen con dos regiones similares sobre un fondo de diferente intensidad. El histograma sólo muestra dos picos, por lo que las dos regiones se "ven" como una. En la parte central, se muestra un imagen de sombreado que se aplica a la imagen con dos regiones. Esto ocaciona, como se ilustra en la parte inferior, que ya no sean distinguibles las regiones en el histograma.

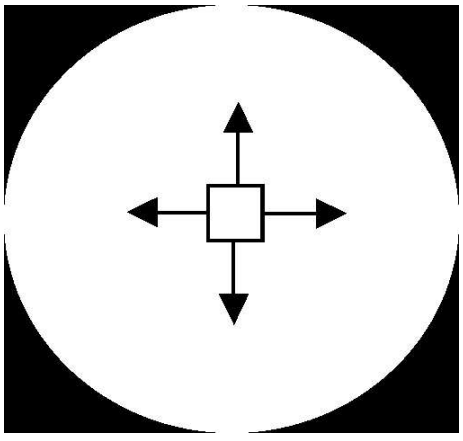


Figura 8.7: Crecimiento de regiones: A partir de una posición inicial se "crece" hasta encontrar una discontinuidad.

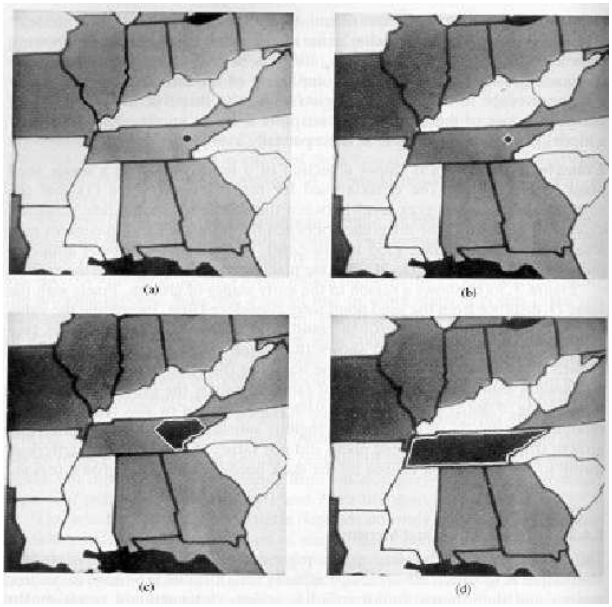


Figura 8.8: Ejemplo de creciemiento de regiones. En cada imagen, (a) – (d), se ilustran diferentes etapas del proceso de crecimiento de una de las regiones (estado) de la imagen del mapa de los E.U.A.

espacio de estados, donde los estados son todas las posibles particiones, para encontrar la mejor partición.

Una forma de realizar esta búsqueda es utilizando información de orillas. Se toman inicialmente las orillas obtenidas por cierto detector de orillas, incluyendo su magnitud y dirección. Entonces se eliminan orillas de acuerdo a ciertos criterios, formando regiones mayores. Los criterios se basan en alguna de las siguientes consideraciones:

- Eliminar la orilla si su magnitud es menor a cierto límite.
- Eliminar la orilla si no existen orillas contiguas de dirección similar.
- Eliminar la orilla si la diferencia entre niveles de gris entre las regiones que separa es menor a cierto límite.
- Eliminar orillas cuando el perímetro de la región que separan es menor a cierto límite.

De esta forma, se van eliminando regiones "no significativas", quedando aquellas que, en principio, representan partes u objetos de la imagen. Un ejemplo del proceso de crecimiento de regiones por eliminación de orillas se muestra en la figura 8.9, donde alguna orillas se han eliminado de una etapa a la siguiente.

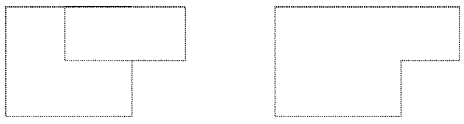


Figura 8.9: Ilustración del proceso de crecimiento de regiones por eliminación de orillas. En la imagen izquierda, se muestran las orillas detectadas en una imagen, las cuales delimitan dos regiones. En la imagen derecha, se han eliminado algunas orillas (en la parte central), quedando solamente una región.

Esta técnica se puede combinar con loa métodos de agrupamiento de orillas (del capítulo anterior), de forma que las orillas que se mentangan al final del proceso se agrupen en bordes, delimitando las regiones significativas.

8.3.2 Técnicas basadas en grafos

Las técnicas basadas en grafos se basan en una representación gráfica de las regiones y sus relaciones (regiones vecinas) denominada grafo de vecindad de regiones (*region adjacency graph*).

En esta representación gráfica, los nodos representan regiones y los arcos relaciones con otras regiones vecinas. Este grafo debe ser planar. Para simplificar el análisis, se agrega una región "virtual" que rodea a la imagen, y a la cual también se le asocia un nodo. Las regiones que delimitan con la orilla de la imagen, se conectan a esta región virtual. También se puede obtener el dual del grafo, insertando nodos en cada "región" del grafo (que normalmente corresponden a vértices en la segmentación original de la imagen) y uniéndolos por arcos que cruzan cada arco del grafo original. En este grafo dual, los arco corresponden a los contornos (orillas) y los nodos a donde se unen 3 o más segmentos de contorno. En la figura 8.10 se muestra el grafo de vecindad y el grafo dual para una imagen sencilla.

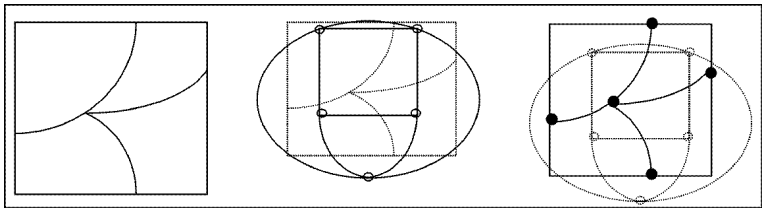


Figura 8.10: Grafos de vecindad de regiones. De izquierda a derecha: (a) imagen con 4 regiones, (b) grafo de vecindad, (c) grafo dual.

Esta representación es muy útil para implementar algoritmos para agrupamiento de regiones, manteniendo ambos grafo actualizados. Dada una segmentación inicial, por ejemplo en base a orillas, se utiliza el grafo de vecindad y su dual para el agrupamiento de regiones. Para agrupar dos regiones vecinas, R_i, R_j, el algoritmo es el siguiente:

1. Grafo de vecindad:
   (a) Poner arcos entre R_i y todos los vecinos de R_j, si no estan conectados.
   (b) Eliminar R_j y sus arcos asociados.
2. Grafo dual:
   (a) Eliminar los arcos que corresponden a los bordes entre R_i y R_j.
   (b) Para cada nodo conectado por los arcos eliminados:
      i. Si el grado del nodo es 2 (tiene 2 arcos), eliminar el nodo y convertirlo en un sólo arco.
      ii. Si el grado del nodo es mayor o igual a 3, actualizar las etiquetas de los arcos que estaban conectados a R_j.

8.4 Segmentación por división-agrupamiento

En la segmentación por división–agrupamiento (*split & merge*), se combinan técnicas globales (división) y globales (agrupamiento). Normalmente se parte de una segmentación inicial, obtenida mediante orillas o regiones, a partir de la cual se agrupan o dividen regiones. Est se facilita utilizando una representacion basada en estructura piramidal y/o en árboles cuaternarios.

A continuación veremos una técnica de segmentación por división–agrupamiento mediante una representación priramidal, y después una extensión con árboles cuaternarios.

8.4.1 Método basado en pirámide

Inicialmente la imágen se estructura en una forma piramidal, de forma que cada posible región (cuadrante) está formada de 4 subregiones. Para ello se representa la imagen a diferentes resoluciones, desde cierta resolución máxima hasta cierta resolución mínima[1]. Entonces se aplica el siguiente algoritmo:

1. Considerar una medida de homogenidad: H (diferencia en niveles de gris, por ejemplo). A partir de cierto nivel (arbitrario) de la pirámide:
   (a) División. Si una región no satisface la medida (H falso), dividirla en 4 regiones, continunado a los niveles inferiores de la pirámide.
   (b) Agrupamiento. Si 4 regiones contiguas (mismo cuadrante) satisfacen el criterio (H verdadera), agruparlas en una región y continuar a los niveles superiores de la pirámide.
2. Una vez segmentada la imagen en cuadrantes en cada nivel, realizar un agrupamiento de regiones vecinas a diferentes niveles. Si hay dos regiones contiguas, al mismo o diferente nivel, tal que $H(R_i \cup R_j) =$ verdadero, agruparlas en una región.
3. Si no hay más regiones que dividir o agrupar, terminar.

Al final del proceso, se obtiene una segmentación en N regiones contiguas de la imagen. La suavidad en la aproximación de la forma de las regiones mediante cuadrantes, depende de la resolución máxima utilizada en la pirámide. La figura 8.11 ilustra esta técnica para una imagen sintética con dos regiones.

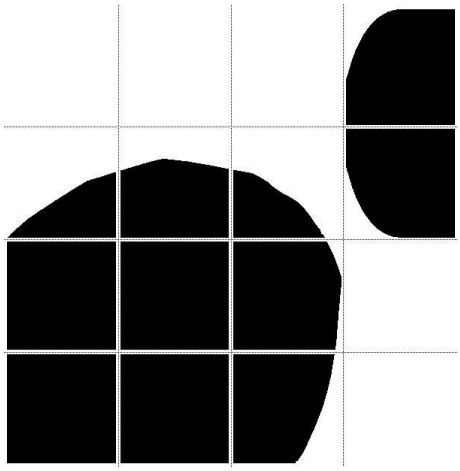


Figura 8.11: Ejemplo de segementación por división–agrupamiento en una imagen sintética. Se ilustran con líneas punteadas los cuadrantes a la máxima resolución de la pirámide, y separadas con líneas continuas, las 3 regiones detectadas.

8.4.2 Método basado en árboles cuaternarios

Una variente del método de división de agrupamiento es una técnica basada en árboles cuaternarios (*Quadtrees*) que se utiliza cuando se desea segmentar cierta región en particular.

Se considera una región de interés que se desea segmentar. Para ello se toma un nivel esperado de gris (I) de dicha región, el cual puede ser estimado del histograma. Las medidas de cercanía

[1] Ver *Pirámides y árboles cuaternarios* en el Capítulo 8.

al nivel esperado y de homegeneidad de la región, se basan en el uso de las estadísticas siguientes: promedio (μ) y desviación estandar (σ). Se utilizan dos constantes:

k_1: tolerancia para el nivel de gris respecto al nivel esperado.

k_2: tolerancia para la homegenieidad de la región.

En base a estos parámetros, se definen las siguientes relaciones:

En rango:

$$I - k_1 < \mu < I + k_1 \tag{8.1}$$

Menor al rango:

$$\mu \leq I - k_1 \tag{8.2}$$

Mayor al rango:

$$\mu \geq I + k_1 \tag{8.3}$$

Uniforme:

$$\sigma < k_2 \tag{8.4}$$

No unifrome:

$$\sigma \geq k_2 \tag{8.5}$$

Las que permiten especificar 3 tipos de regiones:

- *Región uniforme en rango*: satisface 8.1 y 8.4
- *Región uniforme fuera de rango*: satisface [8.2 o 8.3] y 8.4
- *Región no uniforme*: satisface 8.5

Entonces el algoritmo de segmentación para una región es el siguiente:

1. Dividir la imagen en 4 y calcular la media (μ) y desviación estandar (σ) de la intensidad en cada cuadrante.
2. Si es una *región uniforme en rango*, tomar dicho cuadrante como una región base y pasar al paso 5.
3. Si es una *región uniforme fuera de rango*, desechar dicho cuadrante.
4. Si es una *región no uniforme*, entonces dividir el cuadrante en 4 y repetir (1) a (3) hasta que todos los cuadrantes sean región base o estén fuera de rango, o ya no sea posible dividirlos.
5. Tomar como la región buscada el cuadrante mayor y unirle todos los cuadrantes adyacentes que satisfagan la condición de *región uniforme en rango*.

La figura 8.12 ilustra la aplicación de esta técnica en una imagen de endoscopía en la que se busca la región “obscura”.

Esta técnica puede extenderse a varias regiones tomando varios valores base y aplicando en forma análoga el algoritmo anterior para varias regiones, considerando el promedio y desviación estandar por región.

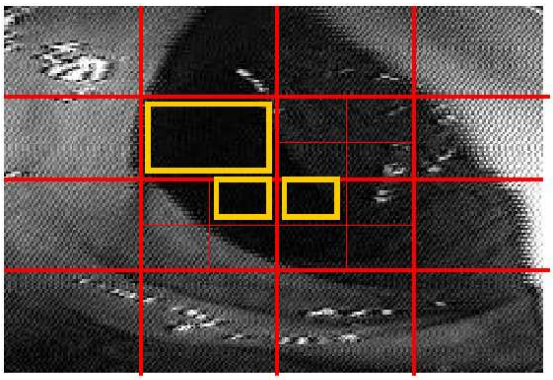


Figura 8.12: Ejemplo de segmentación mediante árboles cuaternarios. Se ilustran sobre la imagen las divisiones, y las regiones base (cuadros) que corresponden a región "obscura" en una imagen de endoscopía.

8.5 Incorporación de semántica del dominio

Las técnicas anteriores tienen sus limitaciones. Todas están basadas en heurísticas genéricas y no toman en cuenta las posibles interpretaciones que pudiera tener cada región. Una forma de tratar de mejorar dichas técnicas es utilizando la semántica del dominio; es decir, información *a priori* de la clase de objetos que esperamos en las imágenes.

Para incoporar semántica del dominio en la segmentación, se realiza una segmentación inicial basada en las técnicas genéricas y luego se busca mejorarla mediante la incorporación de semántica. Para ello se consideran varios posibles parámetros de cada región:

- dimensiones de la región y el contorno,
- forma de la región,
- posición de la región en la imagen,
- posición de la región respecto a otras regiones.

Para esto se puede tomar un enfoque bayesiano considerando las probabilidades de que cada región R_i tenga cierta interpretación X dadas sus mediciones; y de que exista una separación B_{ij} entre dos regiones R_i y R_j dadas las mediciones del contorno:

$$P(R_i - es - X \mid mediciones - de - R_i) \tag{8.6}$$

$$P(B_{ij} - entre - R_i - y - R_j \mid mediciones - de - B_{ij}) \tag{8.7}$$

Por ejemplo, si consideramos que se van a segementar imágenes de paisajes, las posibles interpretaciones para las regiones son cielo, arbustos, pasto, bosque, etc. A cada clase de región se la asocian ciertas probabilidades en base a atributos como color, posición espacial en la imagen, entre otros. De esta forma, para regiones de tipo *cielo*, se puede especificar:

$$P(R_i - es - Cielo \mid color - de - R_i), \tag{8.8}$$

con valores altos de probabilidad para colores en el rango de los "azules", y baja probabilidad si esta fuera de este rango.

Si se tienen varios atributos por región, éstos se combinan mediante teoría de probabilidad. Normalmente es más fácil estimar a partir de datos (regiones conocidas), la probabilidad inversa, es decir, $P(mediciones - de - R_i \mid R_i - es - X)$. La probabilidad de que cada región pertenezca a cierto tipo, puede ser entonces estimada por el teorema de Bayes:

$$\begin{gathered} P(R_i - es - X \mid mediciones - de - R_i) = \\ \frac{P(R_i - es - X)P(mediciones - de - R_i \mid R_i - es - X)}{P(mediciones - de - R_i)} \end{gathered} \tag{8.9}$$

$P(R_i - es - X)$ es la probabilidad a priori (que se pueden considerar iguales), $P(mediciones - de - R_i \mid R_i - es - X)$ se puede obtener como el producto de las probabilidades de cada atributo considerando independencia, y $P(mediciones - de - R_i)$ es una constante (K) de normalización (no es necesario obtenerla). Si, por ejemplo, se estima la probabilidad de que sea "cielo" para una región, dados dos atributos (color y posición), se aplica la siguiente expresión:

$$\begin{aligned} P(R_i - es - Cielo \mid color - de - R_i, pos - de - R_i) = \\ K \times P(R_i - es - Cielo)P(color - de - R_i \mid R_i - es - cielo) \\ P(pos. - de - R_i \mid R_i - es - cielo) \end{aligned} \tag{8.10}$$

Dada una segmentación inicial obtenida por algun método genérico, el procediamiento de mejora mediante segmentación semántica es:

1. Asignar una interpretación inicial a cada región en base a las probabilidades de los atributos considerados.
2. Agrupar:
   (a) regiones contiguas con la misma interpretación,
   (b) regiones contiguas para las cuales el contorno que las separa tenga baja probabilidad.
3. Re–evaluar las probabilidades de cada región.
4. Repetir hasta que no existan regiones por agrupar.

La figura 8.13 ilustra la idea de la segmentación semántica en una imagen sintética, en la cual dos regiones que corresponden a "cielo" se agrupan ya que tienen la misma interpretación.

Las probabilidades requeridas se obtienen en base a estadísticas de imágenes del dominio segmentadas correctamente.

8.6 Sistema experto para segmentación

Extendiendo la idea de utilizar semántica para segmentación, se han desarrollado sistemas de segmentación más sofisticados basados en una representación explícita del conocimiento del dominio. Este tipo de sistemas se conoce como *sistemas basados en conocimiento* o *sistemas expertos.*

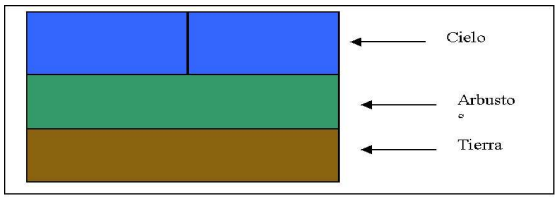


Figura 8.13: Segmentación semántica. El cielo está dividido en 2 regiones; las cuales se unen en base a información semántica.

Un sistema experto utiliza una representación explícita del conocimiento y técnicas de razonamiento simbólico para resolver problemas en un dominio específico. El conocimiento generalmente se representa por medio de reglas "Si *condiciones* Entonces *conclusión*" que se almacenan en la *Base de Conocimientos* y que operan sobre los datos y conclusiones almacendas en la *Memoria de Trabajo* mediante una *Máquina de Inferencia*. En la figura 8.14 se muestra la arquitectura básica de un sistema experto con sus partes principales.

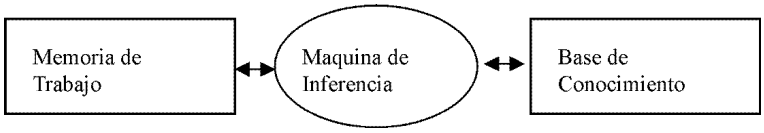


Figura 8.14: Arquitectura básica de un sistema experto. El conocimiento que reside en la base de conocimiento es aplicado mediante la máquina de inferencia a los datos en la memoria de trabajo.

Nazif y Levine desarrollaron un sistema experto para la segmentación de imágenes, el cual pretende ser "genérico"; es decir, aplicable a cualquier tipo de imagen. Para ello combinan conocimiento de técnicas de segmentación y del dominio (semántica) para lograr un sistema de segmentación más flexible y robusto. El sistema tiene 3 tipos de reglas agrupadas en 3 niveles jerárquicos:

1. Reglas de segmentación. Contiene reglas para análisis de regiones, análisis de líneas y análisis de reglas.

2. Reglas de foco de atención. Seleccionan el área de la imagen a ser analizado primero.

3. Reglas de estrategia. Seleccionan las reglas de foco de atención (estrategia) más adecuada de acuerdo al tipo de imagen.

Los dos niveles superiores son reglas de control que deciden la ejecución de otras reglas (meta–reglas).

Ejemplos de reglas de nivel I:

Si	la dimensión de la región es muy baja,
	la vecindad con otra región es alta,
	la diferencia en atributo-1 con otra región es baja
Entonces	agrupa las dos regiones

Si la varianza de la región es alta,
el histograma de la región es bimodal
Entonces divide la región

Ejemplos de reglas de nivel II y III:

Si el gradiente de la línea es alto,
la longitud de la línea es grande,
existe la misma región a ambos lados de la línea,
Entonces analiza dichas regiones

Si el gradiente promedio de la región es alto,
la dimensión de la región es grande,
Entonces obten la línea que intersecta la región

El sistema también incluye reglas para la iniciar y terminar el proceso.

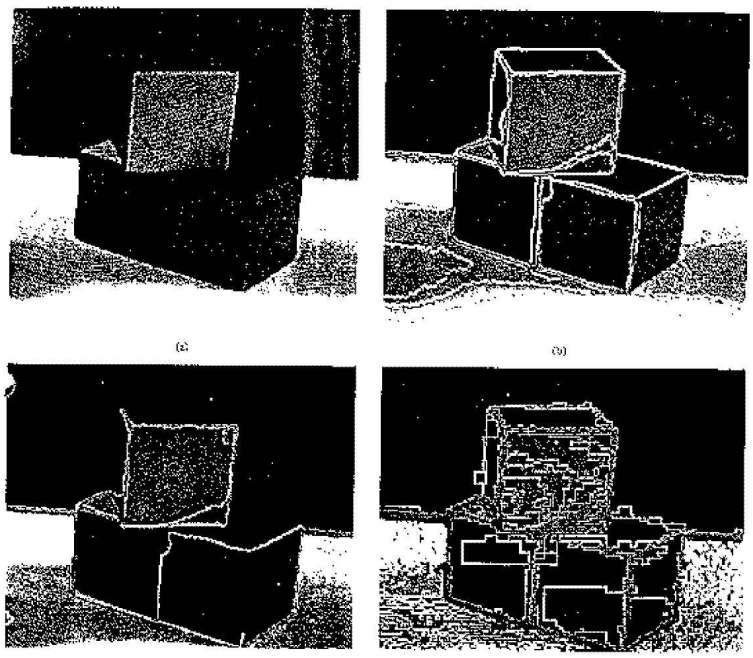


Figura 8.15: Ejemplo de segmentación de una imagen utilizando el sistema experto. En la parte superior se muestra la imagen original (izq.) y el resultado de la segmentación (der.) con el sistema experto. Las imágenes en la parte inferior presentan el resultado con dos técnicas de segmentación, que son inferiores a los del sistema experto.

La imagen se almacena en la memoria de trabajo, y se van aplicando las reglas, utilizando la máquina de inferencia, almacenando los resultados parciales en la misma memoria de trabajo. El proceso termina cuando ya no existan reglas que se puedan aplicar o hasta llegar a las condiciones de terminación. La figura 8.15 muestra la aplicación del sistema experto de segmentación a una imagen, comparandola con otras técnicas.

8.7 Referencias

La segmentación de imágenes es una de las áreas más importantes y complejas en visión, la cual ha sido estudiada extensamente y continua siendo tema de descusión. En su forma más basica, segmentación consiste en *etiquetar* los pixeles como pertenecientes a una clase. Por ejemplo, si se toman sólo dos clases, objeto y fondo, el etiquetado sería de la forma $2^{largo*alto}$, lo cual puede ser visto de dos maneras (1) existe una solución unica, entonces el problema tiene una complejidad $NP - Completo$, (2) existen múltiples soluciones por lo que el problema se transforma en mal-planteado (*ill-posed*) y es necesario asumir restricciones para seleccionar alguna de estas múltiples soluciones. Prácticamente todos los algoritmos de segmentación caen en el segundo apartado, se utiliza información del dominio.

La restriccion más sencilla que se asume es que las características varian suavemente a través de la imagen. La justificación parte de que regiones uniformes y contiguas *usualmente* corresponden a superficies de un mismo objeto físico. Otros trabajos han utilizado semántica del dominio en el sentido de calcular las restricciones geométricas. Uno de los primeros trabajos en utilizar esta suavidad local, para segmentar crecimiento regiones, fue el de Brice y Fennema [7] quienes reconocían triángulos y cuadrados.

Una de las primeras referencias de segmentación por histograma es el presentado por J. Prewitt [93] (en este mismo capítulo se presenta su detector de orillas). Los histogramas manejados ese trabajo eran bimodales. Como es de esperarse para imagenes reales no es siempre posible encontrar dos picos en el histograma. Una mejora al trabajo de Prewitt es el de Chow y Kaneko [15] quienes dividen la imagen hasta encontrar regiones bimidales para después umbralizar. Este tipo de técnica se les conoce como *multi threshold*. En [84] se presenta una comparación entre quince algoritmos de binarización; aun cuando ninguno ganó claramente, la técnica de Niblack [57] se comportó mejor en la mayoría de los casos. Información general de métodos de segmentación pueden encontrarse en Ballard [2] (cap. 5).

El crecimiento de regiones es una idea que se sigue utilizando pero no sólo basado en la media o desviación estándar local. Por ejemplo, fuzzy y dimensión fractal [54], magnitud del gradiente [101], grafos (cortes normalizados) [106].

Una excelente referencia para árboles cuaternarios puede encontrarse en [116]. El método de segmentación basado en árboles cuaternarios se describe en [59]. Técnicas recientes como el *hyper-stack* [132] utilizan quadtrees y multiescala para segmentación.

Para mayor infromación sobre el sistema experto para segmentación, consultar [82].

8.8 Problemas

1. ¿En qué consiste la incorporación de semántica del dominio en segmentación? ¿Qué ventajas y desventajas tiene respecto a técnicas que no consideran conocimiento del dominio?

2. Dado un histograma con ruido (picos y valles "falsos"), plantea un algoritmo para detectar los picos y valles "significativos" para segmentar la imagen en "N" regiones.

3. Segmenta la imagen de la figura 8.11 utilizando la técnica de división-agrupamiento, mostrando el desarrollo.

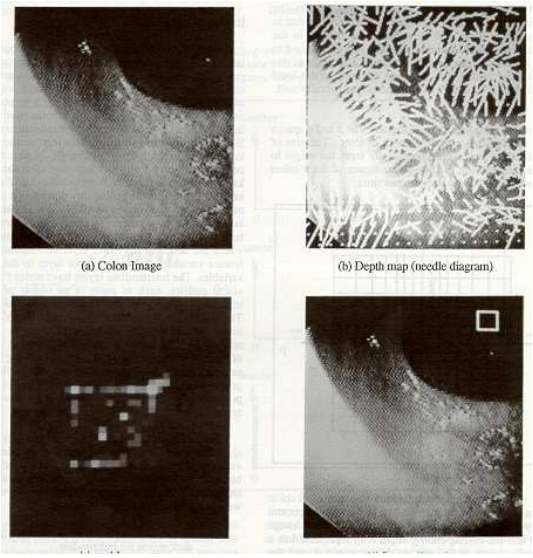
4. Para la técnica de segmentación por *Quadtrees* indica la forma de la estructura (ligas) que la haría más eficiente.

5. Se desea extraer la región "más brillante" de la imagen y que sea de cierto tamaño mínimo. Plantea un algoritmo que combine la técnica de histograma y árboles cuaternarios para lograr esto.

6. Dadas imágenes de polígonos regulares, como tríangulos, rectángulos, pentágonos, plantea métodos para segmentar los polígonos: (a) global, (c) local, (c) división–agrupamiento. Muestra el desarrollo, sobre una imagen sintética, de cada método.

Capítulo 9

Movimiento



9.1 Introducción

El análisis de imágenes en movimiento es en cierta forma análogo al problema de estéreo. Consiste en integrar la información de dos o más imágenes con pequeñas diferencias espaciales para ayudar a su interpretación. Además de obtener información del movimiento de los objetos o del observador, se facilita el obtener otra información, como la tercera dimensión (forma de movimiento), la segmentación y el reconocimiento.

Al considerar movimiento, puede ser que los objetos se muevan, o que la cámara se mueva, o ambos. Sin embargo, todos los casos se pueden agrupar en uno considerando que existe un movimiento relativo entre cámara y objetos. De esta forma se obtiene una secuencia de imágenes entre las que existen pequeñas diferencias debidas a dicho movimiento relativo. En la figura 9.1 se ilustra un ejemplo de movimiento relativo, en el cual el objeto (círculo) en la imagen va aumentando de tamaño. Esto puede ser debido a que el objeto se acerca a la cámara, el observador se acerca al objeto, o ambos.

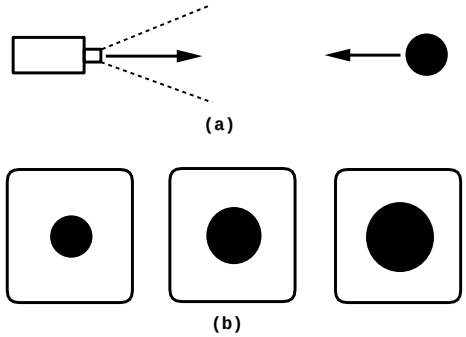


Figura 9.1: Movimiento Relativo. En (a) se ilustra el movimiento relativo del observador y del objeto, que se están acercando. Esto tiene el efecto que el objeto aumente de tamaño en la imagen, como se ilustra en la secuencia de imágenes en (b).

El análisis de una secuencia de imágenes se puede ver desde dos puntos de vista, que corresponden a los dos enfoques principales para imágenes en movimiento:

- *Continuo* - se considera la secuencia de imágenes como un flujo de intensidades combiantes a lo que se denomina *flujo óptico*.
- *Discreto* - se considera la secuencia de imágenes como un conjunto de diferentes imágenes estáticas.

A continuación veremos el primer enfoque, el de flujo óptico y después el de múltiples imágenes.

9.2 Flujo óptico

Diversos experimentos han demostrado que la vista humana responde directamente al movimiento, que se puede considerar como uno de los aspectos básicos de la visión humana. Para esto se considera que el movimiento produce cambios diferenciales en la imágen que son percibidos como una especie de flujo de pixels en el espacio. Esto se puede ver como un arreglo de vectores, cada uno expresándo la velocidad instantánea de cada punto. A dicho arreglo de vectores de velocidad se le denomina el flujo óptico y puede ser obtenido de la secuencia de imágenes. En la figura 9.2 se ilustra este fenómeno. Del lado izquierdo se muestra una imagen, para la cual se muestra el flujo óptico (como vectores) del lado derecho, considerando que el observador se esta acercando a los objetos

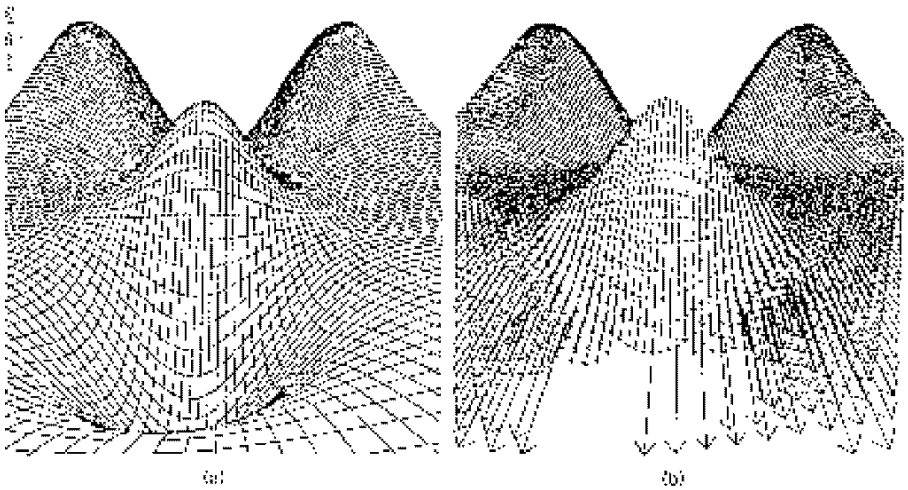


Figura 9.2: Flujo óptico. En (a) se muestra una imagen sintética. Si se considera que el observador se mueve acercándose a los objetos, se obtiene el flujo óptico que se ilustra en (b) con *vectores*.

A partir del flujo óptico se pueden obtener otras características, como el movimiento global, información de profundidad (3-D) y orillas.

9.2.1 Obtención del flujo óptico

Para estimar el flujo óptico, se considera a la secuencia de imágenes en movimiento como un función continua en 3 dimensiones, x, y, y tiempo (t): $f(x, y, t)$. Es decir, al integrar la secuencia de imágnes en diferentes tiempos, la intensidad de cada punto depende de su posición en la imagen (x, y), y de la imagen en el tiempo, t. Esto se ilustra en la figura 9.3

Si consideramos un cambio diferencial en cualquier de las 3 coordenadas, podemos aproximarla mediante su expansión en series de Taylor:

$$f(x + dx, y + dy, t + dt) = f(x, y, t) + \frac{df}{dx}dx + \frac{df}{dy}dy + \frac{df}{dt}dt + TOS \tag{9.1}$$

Donde despreciamos los términos de orden superior (TOS). Si consideramos un cambio muy pequeño, entonces podemos decir que las imágenes son casi iguales:

$$f(x + dx, y + dy, t + dt) = f(x, y, t) \tag{9.2}$$

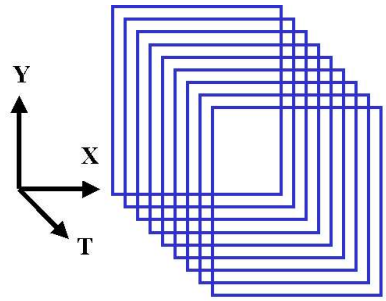


Figura 9.3: Secuencia de imágenes.

Igualándo las dos ecuaciones anteriores, eliminándo los términos comunes y dividiendo entre dt obtenemos:

$$-\frac{df}{dt} = \frac{df}{dx}\frac{dx}{dt} + \frac{df}{dy}\frac{dy}{dt} \tag{9.3}$$

Donde:

$$\mathbf{u} = (u, v) = (\frac{dx}{dt}, \frac{dy}{dt}), \tag{9.4}$$

es el vector de velocidad instantánea que buscamos. Las demás son cantidades medibles de las imágenes (cambios respecto a x, y, t). Si denotamos al gradiente espacial $\Delta f = (\frac{df}{dx}, \frac{df}{dy})$ obtenemos la siguiente relación:

$$-\frac{df}{dt} = \Delta f \cdot \mathbf{u} \tag{9.5}$$

Esta ecuación limita el flujo óptico pero no lo determina. Para calcular **u** utilizamos una técnica iterativa basada en relajación.

Para ello se considera que el movimiento es suave y se define un error en términos de derivadas parciales cuadráticas que se busca minimizar. En base a esto se define un método iterativo que disminuye el error en cada iteración hasta que sea menor a cierto valor preestablecido.

9.2.2 Utilización de flujo óptico

Una vez estimado el flujo óptico, éste puede ser utilizado para obtener información adicional de las imágenes. Se aplica para obtener información de profundidad (forma de movimiento) mediante el cálculo del foco de expansión. También se puede utilizar como base para la detección de bordes y la segmentación.

Foco de Expansión (FOE)

Si consideramos que el observador se mueve y los objetos son estáticos, todos los vectores de velocidad parecen unirse en un punto (interior o exterior) de la imagen. A dicho punto se le denomina el Foco de Expansión o *FOE*. Si existen varios objetos con diferentes movimientos, a cada uno corresponde un foco de expnsión. Esto se ilustra en la figura 9.4, donde se tienen dos

FOE: el del objeto (DEMON) que se mueve hacia el frente y hacia la derecha; y el del resto de la imagen que corresponde al movimiento del observador.

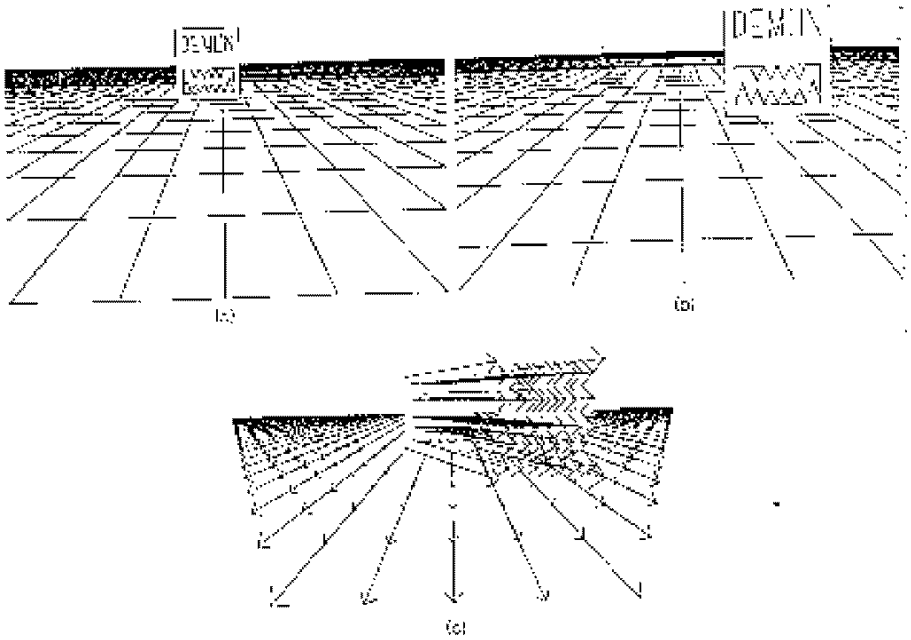


Figura 9.4: Foco de Expansión. En (a) se muestra la imagen en el tiempo incial y en (b) un tiempo después. Los vectores de flujo óptico se ilustran en (c). Si se continuan los vectores del "DEMON" hacia atrás, todos parten de un punto común que es el FOE. Lo mismo sucede para los vectores de "piso" de la imagen.

Para estimar el FOE se hace lo siguiente. Si consideramos proyección prespectiva, y un punto en la imagen en movimiento después de un tiempo t:

$$(x', y') = (\frac{x0 + ut}{z0 + wt}, \frac{y0 + vt}{z0 + wt}) \tag{9.6}$$

Donde w es dz/dt. Si consideramos que t tiende a menos infinito obtenemos el FOE:

$$FOE = (\frac{u}{w}, \frac{v}{w}) \tag{9.7}$$

Profundidad

Existe una relación entre la profundidad (z) de un punto y su distancia al FOE (D), llamada la relación de *tiempo-a-adyacente*:

$$\frac{D(t)}{V(t)} = \frac{z(t)}{w(t)} \tag{9.8}$$

Donde $V(t)$ es la derivada de D respecto a t. Dada la profundidad de un punto podemos obtener la de todos los otros a la misma velocidad:

$$z_2(t) = \frac{z1(t)D2(t)V1(t)}{V2(t)D1(t)} \tag{9.9}$$

Profundidad relativa y orillas

A partir del flujo óptico tambien es posible estimar la profundidad relativa o gradiente de la superficie y detectar cambios bruscos que corresponden a orillas. Para esto consideramos coordenadas esféricas y la velocidad en términos de dicho sistema de coordenadas. Para el caso especial de sólo movimiento del observador, en dirección z y con velocidad S, obtenemos:

$$d\theta/dt = 0 \tag{9.10}$$

$$d\phi/dt = \frac{Sen\phi}{r} \tag{9.11}$$

A partir de esta ecuación podemos calcular la normal en cada punto de la superficie que corresponde a su profundidad relativa. Las discontinuidades en $d\phi/dt$ corresponden a discontinuidades en la superficie, es decir, a orillas. La ventaja de estas orillas, respecto a orillas obtenidas directamente de la intensidad de la imagen, es que corresponden a cambios de profundidad que representan las fronteras de los objetos o cambios bruscos en su superficie. Por lo tanto, a partir de dichas orillas se puede realizar una mejor segmentación de los objetos en la imagen.

9.3 Múltiples imágenes

Un enfoque alternativo al de flujo óptico que considera la secuencia de imágenes como un continuo, es el considerarlas en forma discreta. En este caso, el movimiento se analiza a partir de un conjunto de imágenes. Al considerar múltiples imágenes "estáticas", el problema principal se convierte en el apareamiento de puntos entre las imágenes. Para simplificarlo, podemos consider que el movimiento entre imágenes consecutivas es "pequeño"; es decir, que imágenes consecutivas son similares. Con estas consideraciones, se aplican las siguientes 5 heurísticas de movimiento:

- Velocidad máxima. Un punto tiene una velocidad máxima V y se mueve una distancia máxima $V \times dt$, donde dt es el tiempo que transcurre entre la toma de una imagen y la siguiente.
- Cambios de velocidad. La velocidad de un punto de una imagen a la siguiente es similar; es decir, existen "pequeños" cambios de velocidad (inercia).
- Movimiento común. Regiones de puntos cercanos en la imagen tienen el mismo movimiento o un movimiento muy similar (objetos rígidos).
- Consistencia. Un punto en una imagen corresponde a un solo punto en la siguiente imagen.
- Movimiento conocido. En ocaciones se tiene conocimiento *a priori* del tipo de movimiento de los objetos y/o del observador (modelo de movimiento).

Esta heurísticas de movimiento se ilustran en forma gráfica en la figura 9.5.

Entonces, el problema de análisis de imágenes en movimiento, se enfoca a anlizar las imágenes individuales, segmentándolas en atributos u objetos relevantes, para a partir de éstos buscar similitudes y diferencias entre imágenes y encontrar información del movimiento.

9.3.1 Flujo de Imágenes discretas

Para estimar el movimiento, primero se obtienen puntos relevantes de cada imagen. Por ejemplo, se obtienen orillas o esquinas, de forma que se reduce considerablemente el número de puntos para los

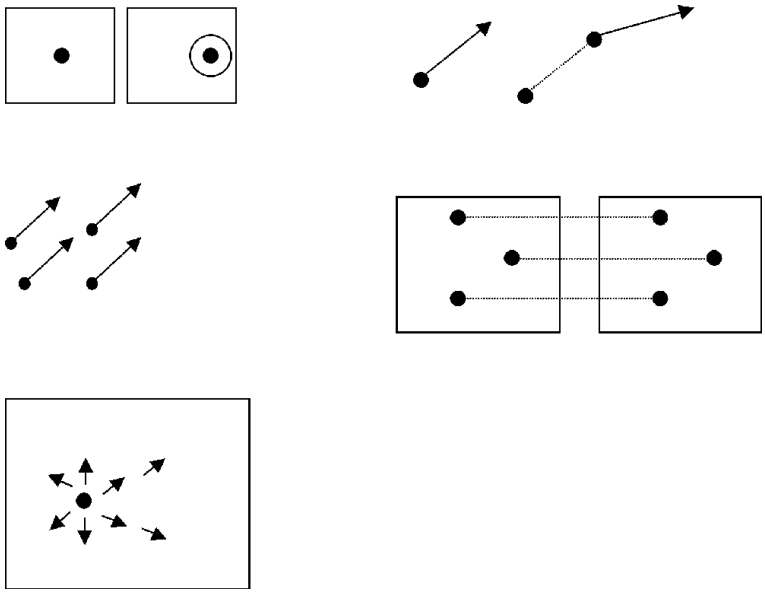


Figura 9.5: Heurísticas de movimiento: (a) distancia máxima, (b) cambios de velocidad, (c) movimiento común, (d) consistencia, (e) movimiento conocido.

que se busca la correspondencia. Depués, se busca la correspondencia entre los puntos relevantes. Para ello se puede aplicar un algoritmo de relajación similar al que se utiliza en estereo. Este se basa en dos heurísticas principales:

- Separación máxima entre puntos correspondientes.
- Puntos cercanos tienen velocidades cercanas.

En la figura 9.6 se muestran dos imágenes, en las que se ha obtenido la correspondencia de las "esquinas".

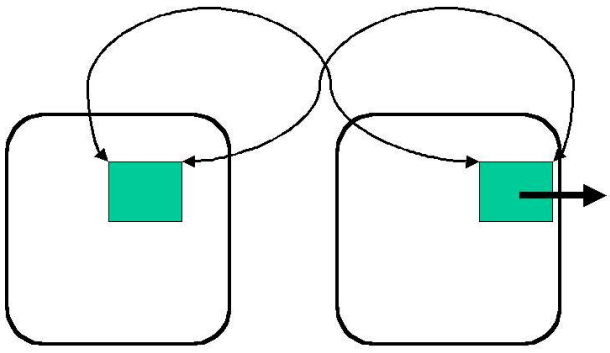


Figura 9.6: Correspondencia. El cuadro de la imagen izquierda se ha desplazado en la imagen derecha, se indica la correspondencia entre las esquinas superiores de ambas imágenes.

Una vez obtenida la correspondencia entre puntos relevantes de imágenes consecutivas, se puede estimar la velocidad en esos puntos. Mediante un proceso de interpolación, se puede extender la estimación de velocidad al resto de la imagen.

Si consideramos que los objetos en la imagen son rígidos (no deformables) y conocemos varios puntos correspondientes entre imágenes en movimiento, podemos usar esta información para es-

timar la forma tridimensional de los objetos. Si tenemos al menos 4 puntos correspondientes, es posible determinar su posición en 3-D (problema análogo al de visión estereoscópica).

9.3.2 Seguimiento

Un uso común del análisis de una secuencia de imágenes en movimiento es el seguimiento (*tracking*) de objetos en las imágenes. Esto tiene múltiples aplicaciones prácticas, como el seguimiento de personas o vehículos para fines de seguridad, el seguimiento de las partes del cuerpo de una persona para reconocer actividades o ademanes, y el seguimiento de vehículos terrestres, marítimos o aéreos en aplicaciones militares.

El seguimiento en una secuencia de imágenes consiste en determinar la posición y velocidad de un punto (o de una región u objeto) en una imagen, dada su posición y velocidad en una secuencia anterior de (una o más) imágenes. El seguimiento se puede realizar en base a diferentes atributos de la imagen, en particular se pueden distinguir las siguientes clases de objetos:

- modelos rígidos bidimensionales o tridimensionales de objetos,
- modelos deformables,
- regiones,
- características de la imagen (puntos, líneas, esquinas, etc.).

Se han desarrollado diversas técnicas para el seguimiento de objetos en imágenes, entre las principales se pueden mencionar:

1. Filtros de Kalman.
2. Técnicas de simulación estocástica, como el algoritmo de condensación.
3. Técnicas heurístcas, que aprovechan las heurśticas de movimiento mencionadas en las sección 9.3.

Dada la localización del objeto o región de interés en una imagen, se pueden utilizar las heurísticas de velocidad máxima y cambios de velocidad para delimitar la región de búsqueda en la siguiente imágen en la secuencia. Esto es particularmente útil si se consideran imágenes con una separación tempral mínima; por ejemplo, 1/30 de segundo. La imagen 9.7 ilustra la aplicación de este principio en el seguimiento de una mano en una secuencia de imágenes. En este ejemplo, la región de la mano es segmentada en base al color de piel, y posteriormente se hace su seguimiento en una ventana alrededor de su posición en la imagen previa, utilizando la heurística de velocidad máxima.

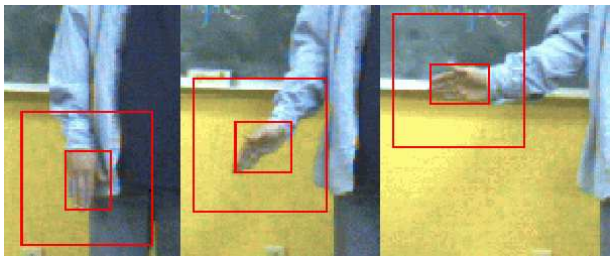


Figura 9.7: Seguimiento. Se ilustra el proceso de seguimiento de una región (en base a color de piel) que se muestra como un cuadro enmarcando la mano de la persona. El cuardo exterior define la región de búsqueda de la mano en la siguiente imagen en la secuencia.

9.4 Navegación

La navegación consiste en poder desplazarse en un cierto ambiente, interior o exterior, en forma segura (sin chocar) y, posiblemente, con una meta o destino final. El análisis de imágenes, en particular de secuencias de imágenes, se han utilizado para navegación. La navegación tiene varias aplicaciones prácticas, entre las que destacan:

- Vehículos autónomos. Navegación de diferentes tipos de vehículos en ambientes exteriores, como carreteras, terrenos e incluso en otros planetas.
- Robots móviles. Navegación de robots principalmente en ambientes interiores como edificios de oficinas, museos, hospitales, etc.
- Robots manipuladores. Navegación de brazos manipuladores en ambientes industriales o en naves espaciales.
- Aplicaciones médicas. Navegación de instrumentos médicos, como endoscopios o laparoscopios.

Existen varias alternativas para la navegación autónoma. Las técnicas dependen del tipo de vehículo, del tipo de ambiente y los sensores con los que se cuente. Se puede utilizar el análisis de imgénes con diferentes técnicas u otro tipo de sensores, como sensores de rango de ultrasonido (sonares) y telémetros laser. Como un ejemplo de la aplicación de visión en navegación, a continuación veremos un enfoque que se basa en el uso de información de profundidad relativa o gradiente. Este algoritmo se orienta a la navegación de robots en interiores o de un endoscopio en el tubo digestivo.

9.4.1 Histograma de Gradiente

Aunque se ha realizado mucha investigación en obtener la forma de objetos a partir de imágenes, no existen muchas aplicaciones prácticas en que se utilize la información tridimensional. Una forma de hacer uso de dicha información es integrándola en un histograma. A esto le denominaremos *Histograma de Gradiente* o *Histograma pq*.

El *histograma de gradiente* consiste en dividir en un número de particiones los valores posibles del gradiente en cada punto (p y q), contando el número de puntos que caigan en cada partición e integrándolos en un histograma bidimensional.

Si consideramos sólo el gradiente respecto a x (p), y que la cámara observa un objeto plano (pared) a un ángulo ϕ, entonces todos los puntos se agrupan en el slot $1/tan\phi$ en el histograma de p. Si la pared es perpendicular al eje de la cámara este valor es 0, y si es paralela, es infinito. La figura 9.8 muestra estos 3 casos. Si extendemos esto al gradiente en ambas direcciones obtenemos un histograma que nos da una indicación de la dirección dominante del entorno que observamos.

Al considerar p y q se obtiene un histograma bidimensional. Para ello se discretizan los valores, de forma de obtener un conjunto finito de rangos, en los cuales se agrupan los vectores de gradiente. Se cuentan los vectores que caen en cada rango y se se *acumulan* en el histograma. Debido a la naturaleza de la tangente, conviene dividir el histograma en forma logarítmica, de forma que se incrementa el tamaño de cada partición en forma exponencial a partir del origen. La figure 9.9 muestra el histograma de gradiente bidimensional.

Una vez obtenido el histograma, se obteine la celda donde se tiene una mayor concentración de vectores de gradiente (el pico del histograma). El valor de pq correspondiente sirve de base para navegación, ya que exite una relación directa de dichos valores con el espacio libre del ambiente por donde se puede navegar. Este algoritmo es útil en particular en ambientes parecidos a un “tubo”, como pasillos en edificios o el tubo digestivo en endoscopía.

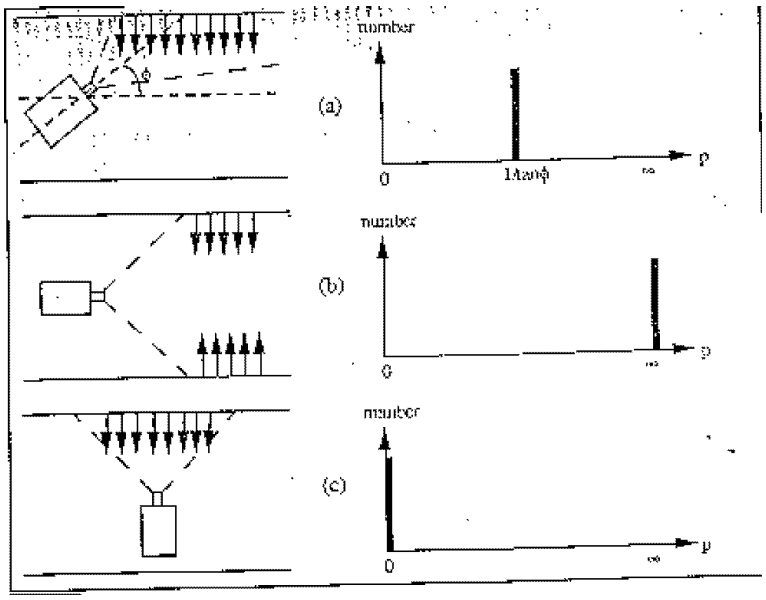


Figura 9.8: Histograma de gradiente en p: (a) pared a un ángulo ϕ, (b) paredes paralelas, (c) pared perpendicular.

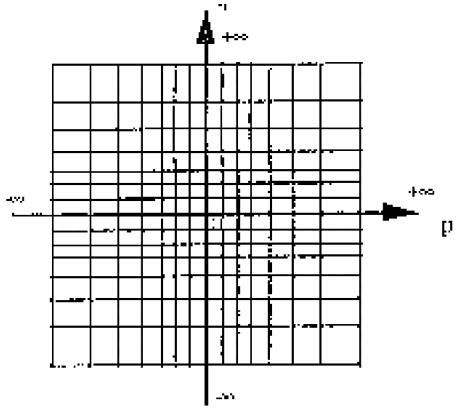


Figura 9.9: Histograma de gradiente bidimensional o histograma pq.

9.4.2 Aplicaciones

El histograma de gradiente ha sido aplicado en navegación para: (i) endoscopía, (ii) robots móviles en interiores.

Endoscopía

Si consideramos que se quiere realizar navegación en un tubo o pasillo angosto (ver figura 9.10), requerimos encontrar el centro para seguirlo.

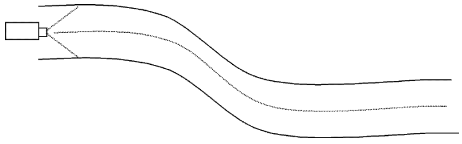


Figura 9.10: Navegación en un tubo.

Aunque en este caso no todos los vectores pq son iguales, si la distancia a las paredes del tubo es relativamente peque'ná en comparación al diámetro, los vectores tienden a agruparse en un *slot*. Este corresponde a la dirección dominante respecto a la cámara y tiene una relación directa con la posición del centro del tubo. De esta forma, obteniendo el "pico" del histograma, podemos estimar la posición del centro, y de esta froma la dirección que se debe seguir en el tubo.

Una aplicación práctica de esta técnica ha sido en la navegación semi–automática de un endoscopio en el tubo digestivo. Esto es con el fin de ayudar a un médico a guiar el endoscopio y permitir que se concentre en su labor de diagnóstico y terapéutica. La figura 9.11 ilustra un ejemplo de la estimación de la dirección de navegación (centro del colón o *lumen*) en endoscopía, basado en el histograma de gradiente.

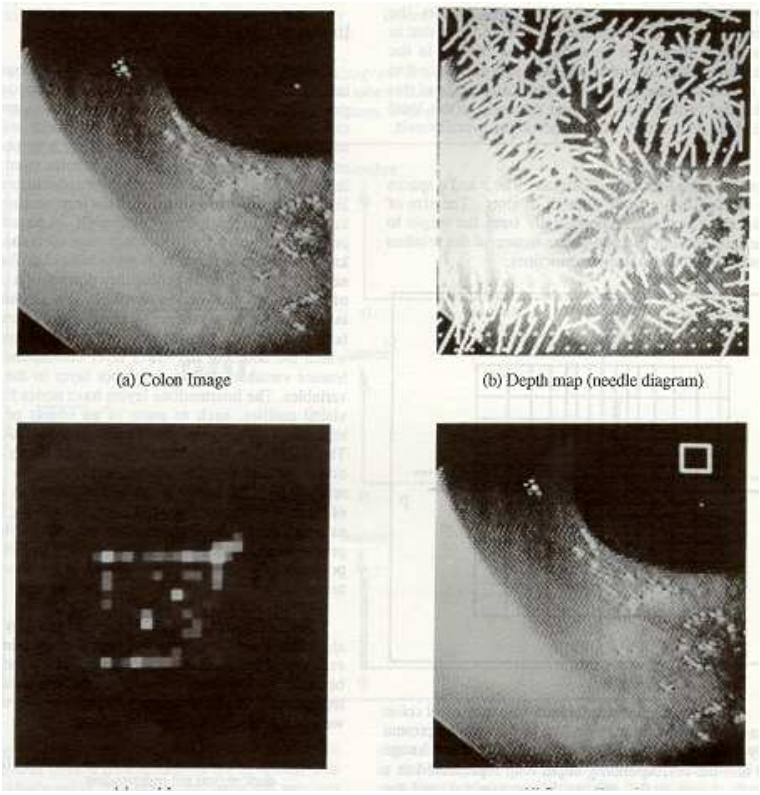


Figura 9.11: Navegación basada en histograma de gradiente en endoscopía. En (a) se muestra la imagen original, en (b) los vectores de gradiente, en (c) el histograma, donde le nivel de gris es proporcional al número de vectores, y en (d) la dirección estimada sobre la imagen original.

Robots móviles en interiores

Otra aplicacion práctica del histograma de gradiente es en la navegación de un robot móvil en interiores (pasillos). En este caso el robot sólo se puede mover sobre el piso (una dimensión), por lo que se considera el histograma respecto a x, el cual se puede obtener sumando los valores por columna (q) para obtener un histograma de gradiente unidimensional. Un ejemplo del uso de este histograma unidimesnional para una imagen de un pasillo se muestra en la figura 9.12.

9.5 Referencias

Existen algunos libros que tratan el análisis de imágenes en movimiento. Ballard [2] en el capítulo 7 aborda el tema de detección de movimiento a partir de flujo óptico y de secuencias de imágenes.

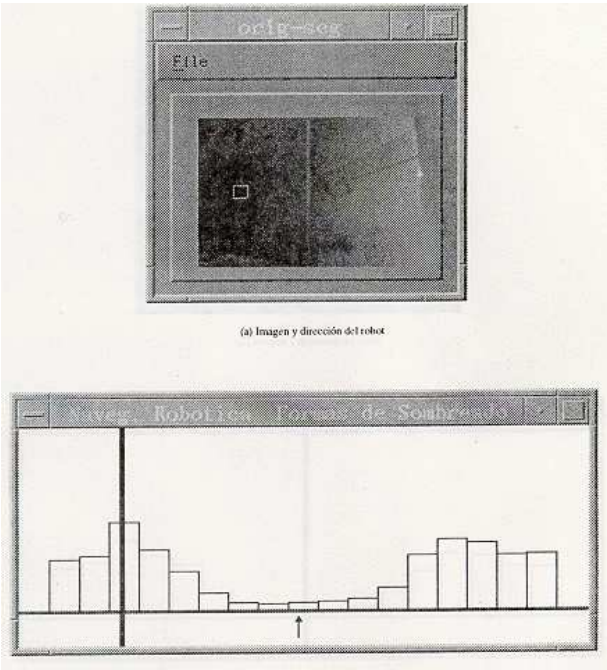


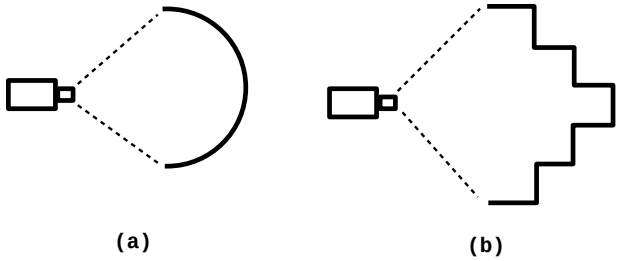
Figura 9.12: Navegación basada en histograma de gradiente en pasillos. En (a) se muestra la imagen de un pasillo con un cuadro sobre puesto que indica la dirección obtenida, en (b) se ilustra el histograma unidimensional indicando el pico mayor.

Faugeras [] trata la determinación de movimiento a partir de puntos y líneas (cap. 7) y de curvas (cap. 9), así como el seguimiento de objetos en imágenes (cap. 8).

La técnica de histograma de gradiente para navegación fue originalmente propuesta por Sucar y Gillies [112] para su aplicación en endoscopía, y luego extendida extendida para robots en interiores por Martínez y Sucar [113].

9.6 Problemas

1. ¿Qué es flujo óptico? ¿Qué ecuación lo limita? ¿Para qué se puede utilizar el flujo óptico?
2. ¿Qué es el gradiente (p,q) relativo de una imagen? ¿De qué forma podemos obtenerlo?
3. Realiza la deducción matemática completa de las ecuaciones de movimiento en coordenadas esféricas.
4. Para los casos (a) y (b) de la figura, obten el histograma *pq* y comenta la información útil que nos da el histograma en cada caso.



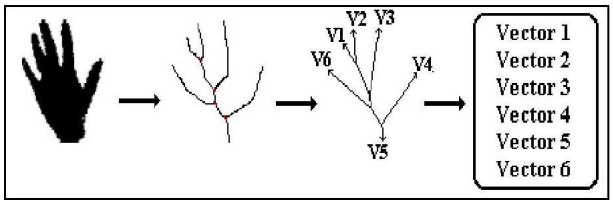
5. Considerando el caso de una fuente lumínica en el mismo punto de la cámara y superficies mate. Obten el valor del gradiente (p, q) para: (a) una imágen de intensidad constante, (b)

una imágen cuya intensidad aumenta linealmente en dirección x. Demuestra que concuerda con los valores obtenidos por el método local.

9.7 Proyectos

1. Implementar en en el laboratorio la obtención del movimiento (vector de velocidad) de un par de imágenes medianet el método de correspondencia. Para ello, utilzando el mismo método de visión esteréo del capítlo anterior, obten la orillas vericales correspondiente entre un par de imágenes (obtenidas de una secuencia en movimimiento). Estima el vector de velocidad para cada orilla y muestra los vectores para la imagen original.

2. Implementar en en el laboratorio el seguimiento de un objeto mediante una técnica heurística. Para ello obten un video de movimiento de un objeto que sea relativamente fácil de identificar (por ejemplo por color o textura). Identifica el objeto en la primera imagen y después haz el seguimiento mediante su búsqueda en una región cercana a la posición en la imagen anterior. Muestra el objeto en la secuencia mediante un rectńgulo que lo identifique en cada imagen.



Capítulo 10

Visión Basada en Modelos

10.1 Visión de alto nivel

Visión de alto nivel busca encontrar una interpretación consistente de las características obtenidas en visión de nivel bajo e intermedio. Se conoce también como *visión sofisticada*. Se utiliza conocimiento específico de cada dominio para refinar la información obtenida de visión de nivel bajo e intermedio, conocida también como *percepción primitiva*. El proceso se ilustra en la figura 10.1. Para esto, se requiere una representación interna o modelo que describe los objetos en el mundo (o en el dominio de interés).

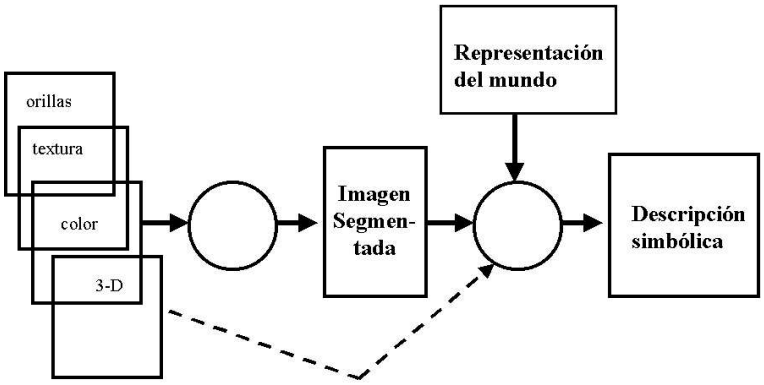


Figura 10.1: Proceso de visión de alto nivel.

Visión de alto nivel tiene que ver, básicamente, con reconocimiento. Es decir, con hacer una correspondencia (*match*) de la representación interna del mundo con la información sensorial obtenida por medio de visión. Por ejemplo, en el reconocimiento de caracteres, se tiene una representación de cada letra en base a ciertos parámetros. Al analizar una imagen, se obtienen parámetros similares y se comparan con los de los modelos. El modelo que tenga una mayor "similitud", se asigna al caracter de la imagen. Una forma de representar caracteres es mediante una codificación radial como se muestra en la figura 10.2

La forma en que representemos tanto los modelos internos como la información de la imagen tiene una gran repercución en la capacidad del sistema de visión.

10.1.1 Representación

Una representación es "un sistema formal para hacer explícitas ciertas características o tipos de información, junto con una especificación de como el sistema hace esto" (definición de David Marr). Hay dos aspectos importantes de una representación para visión:

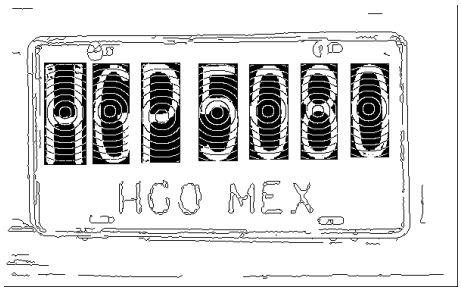


Figura 10.2: Reconocimiento de caracteres en base a su codificación radial.

- Representación del modelo. El tipo de estructura utilizada para modelar la representación interna del mundo.
- Proceso de reconocimiento. La forma en que dicho modelo y la descripción de la imagen(es) son utilizadas para el reconocimiento.

Las representaciones apropiadas para reconocimiento en visión deben de buscar tener las si/-guientes propiedades:

- genéricas;
- eficientes, en espacio y tiempo;
- invariantes, independientes de traslación, rotación y escalamiento;
- robustas, tolerantes a ruido e información incompleta.

Los sistemas de visión de alto nivel se pueden clasificar en dos tipos principales: (i) *Sistemas basados en modelos* que utilizan una representación geométrica (analógica) y el reconocimiento se basa en correspondencia; y (ii) *sistemas basados en conocimiento*, que usan una representación simbólica y el reconocimiento se basa en inferencia.

10.2 Visión basada en modelos

Visión basada en modelos consiste en utilizar una serie de modelos geométricos predefinidos para reconocer los objetos cuya descripción se ha obtenido de la imagen. La estructura general de un sistema de visión basado en modelos se muestra en la figura 10.3. Tiene 3 componentes principales:

- Extracción de características – obtención de información de forma de la imagen para construir una descripción geométrica.
- Modelado – construcción de los modelos geométricos internos de los objetos de interés (*a priori*).
- Correspondenica o *Matching* – apareamiento geométrico de la descripción con el modelo interno.

Los sistemas basados en modelos se pueden dividir en 3 tipos principales:

2-D Utilizan modelos geométricos en 2-D.

2 1/2-D Utilizan cierta información de 3-D como orientación y discontinuidades de la superficie.

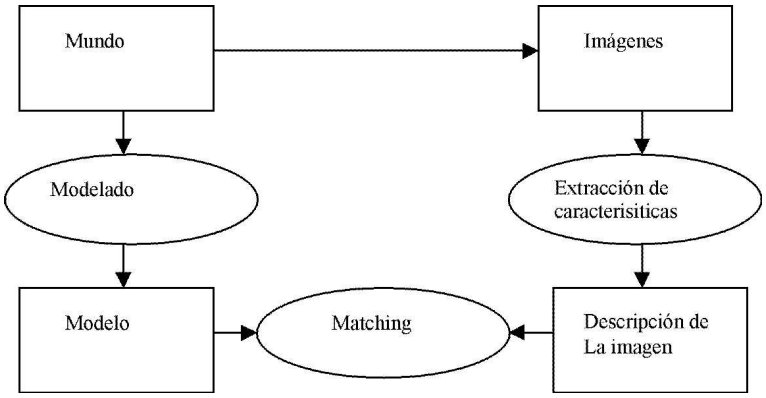


Figura 10.3: Estructura de un sistema de visión basado en modelos.

3-D Representan los objetos en 3-D independientemente del punto de vista.

Las técnicas para correspondencia dependen del tipo de representación. Para modelos que utilizan parámetros globales se utilizan técnicas de reconocimiento estadístico de patrones. Con modelos en base gráficas relacionales, se usan algoritmos de grafos (isomorfismo). En modelos paramétricos se aplican técnicas de optimización paramétrica.

A continuación se analizan diferentes modelos en 2-D y posteriormente en 3-D. Finalmente se verán las técnicas de reconocimiento.

10.3 Modelos en dos dimensiones

Los modelos en dos dimensiones (2-D) están orientados al modelado y reconocimiento de objetos en función de su representación a nivel imagen, es decir, en dos dimensiones. Para representar un objeto en 2-D, existen básicamente dos alternativas:

- Contrornos. El objeto se representa en base a su borde o contorno.
- Regiones. El objeto se representa en base a la región que define.

A continuación veremos varias técnicas para representar objetos en base a contornos y regiones. También se presentan descriptores globales que permiten describir un objeto en base a pocos parámetros.

10.3.1 Contornos

Polilíneas

La representación de *polilíneas* consiste en una descripción de contornos en base a segmentos de línea, donde cada segmento (X) se especifica mediante el punto inicial y final. La concatencación de estos puntos, con el mismo punto inicial y final, describe un contorno:

$$X_1 X_2 ... X_n, X_1 \tag{10.1}$$

Donde X_i corresponde a las coordenadas x, y de cada punto. En la figura 10.4 se muestra un ejemplo de una forma en 2-D representada en pase a polilíneas.

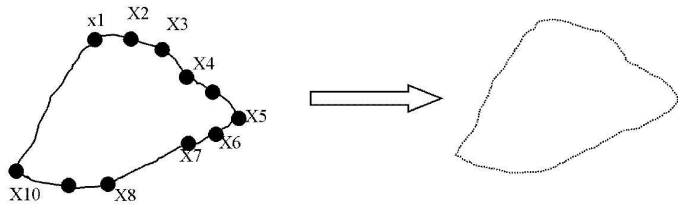


Figura 10.4: Polilíneas. La forma de la izquierda se aproxima mediante una serie de segmentos de recta entre los puntos X_1 a X_{10}, como se muestra a la derecha.

El problema principal es encontar dichos puntos a partir de orillas, o secciones de línea. Una forma de hacerlo es seguir el contorno y comparar la orientación entre orillas vecinas (agrupamiento). Cuando la diferencia se mayor a cierto límite indicar este punto como un punto de quiebre. Otra técnica (división) consiste en aproximar la curva por una línea y calcular la distancia de cada punto a ésta. Si es menor a un límite terminar, sino, poner un punto de quiebre en el punto más lejano y repetir. En la figura 10.5 se ilustran algunos pasos en la obtención de los puntos mediante las técnicas de agrupamiento y división.

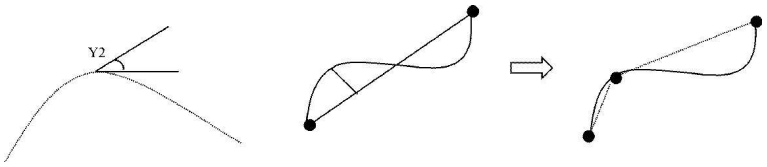


Figura 10.5: Detección de puntos de quiebre. (a) Agrupamiento, cuando la diferencia de orientación es mayor a un cierto ángulo, se marca un punto. (b), (c) División, se se unen los puntos extremos, y se crea un nuevo punto en el punto más alejado de la curva.

Códigos de cadena

Los códigos de cadena consisten, también, de secciones de línea, pero estas están dentro de una retícula fija con un número de orientaciones limitadas (usualmente 4 u 8). Se representa con el punto inicial y la dirección codificada del siguiente segmento. Por ejemplo, si se tienen 4 direcciones básicas, estas se pueden codificar de la siguiente manera (ver figura 10.6-(a)):

0 Izquierda.

1 Arriba.

2 Derecha.

3 Abajo.

La forma de la figura 10.6-(b) tiene el siguiente código bajo esta codificación (considerando que inicia en la esquina superior izquierda, en sentido de las manecillas del reloj):

$$0, 1, 0, 0, 3, 3, 3, 2, 1, 2, 3, 2, 1, 1$$

Se puede obtener la "derivada" del código, la cual consiste en tomar la diferencia entre cada segmento y el segmento anterior, módulo el número (N) de orientaciones:

$$Derivada = [C_i - C_{i-1}]MODN \tag{10.2}$$

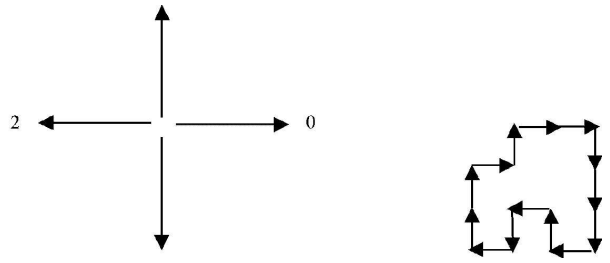


Figura 10.6: Códigos de cadena. En (a) se muestran las 4 direcciones del código básico. En (b) se ilustra un ejemplo de una forma en base a este código.

Para el ejemplo de la figura 10.6 se tiene el siguiente código para la derivada:

$$0, 1, 3, 0, 3, 0, 0, 3, 3, 1, 1, 3, 3, 0$$

Las códigos de cadena tienen varias ventajas sobre polilíneas:

- Representación más compacta en espacio de almacenamiento.
- Más adecuada para realizar reconocimiento independientemente de la posición.
- Su derivada es invariante bajo rotación.
- Son apropiados para el agrupamiento de regiones.

Descriptores de Fourier

Una curva cerrada puede ser representada mediante series de Fourier con una parametrización adecuada. Una parametrización en términos de las componentes en $x(x1), y(x2)$ se muestra en la figura 10.7.

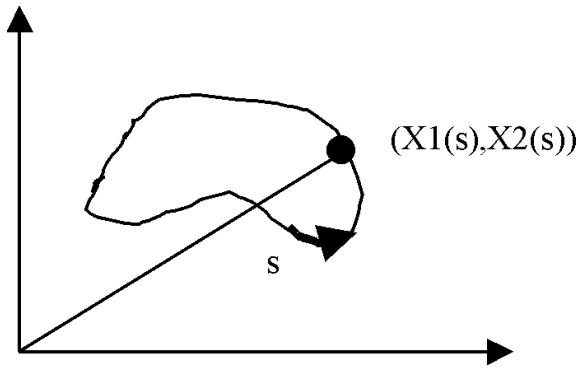


Figura 10.7: Ejemplo de un contorno que se representa mediante descriptores de Fourier

Se considera el contorno esta constituiido por N puntos en el plano x, y. De esta forma el contorno se representa por la serie:

$$X(i) = [x_i, y_i], i = 1..N \tag{10.3}$$

Cada punto se puede considerar como un número complejo:

$$X(i) = x_i + jy_i \tag{10.4}$$

Empezando en un punto arbitrario en el contorno y siguiendolo en alguna dirección (por ej., en contra de las manecillas del reloj) da una secuencia de números complejos. La transformada discerta de Fourier de esta secuencia se obtiene mediante la siguiente expresión:

$$X(s) = \sum X(k) \exp(jk\omega_0 s), s = 0, 1, 2, ..N - 1 \tag{10.5}$$

donde $\omega_0 = 2\pi/N$ Los coeficientes de Fourier están dados por:

$$X(k) = \frac{1}{N} \sum_{0}^{N} X(s) \exp(-jk\omega_0 s) \tag{10.6}$$

De esta forma los descriptores de Fourier pueden representar un contorno cerrado arbitrario. Esta descripción tiene la ventaja que generalmente se logra una buena descripción en base a pocos términos. Además es invariante bajo traslación y rotación.

Secciones cónicas

Los polinomios de grado 2 son adecuados para representar curvas cerradas. Círculos se pueden representar con 3 parámetros, elipses con 5, y cónicas genéricas (círculo, elipse, parábola e hipérbola) con 6. La ecuación general de una cónica es:

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0 \tag{10.7}$$

Esta representación es útil para representar ciertos tipos de objetos, como pueden ser objetos simples para manufactura.

Otra forma de representar curvas es mediante la interpolación aproximada por secciones de curvas. Una representación de este tipo consiste de curvas de grado n conocidas como *B splines*, las cuales son utilizadas ampliamente en gráficas computacionales.

10.3.2 Regiones

Las representaciones anteriores representan un objeto bidimensional en base a su borde o contorno. Otra alternativa es representarlo mediante la región correspondiente. Las siguientes representaciones están orientadas a describir regiones en 2-D:

- arreglos de ocupación espacial,
- eje-Y,
- árboles cuaternarios,
- esqueletos.

A continuación veremos cada una de ellas.

Arreglos de ocupación espacial

Se utiliza un arreglo o predicado de pertenencia $p(x, y)$, que tiene un 1 para cada elemento (pixel) que pertenece a la región y un 0 si no pertenece. La figura 10.8 muestra un ejemplo sencillo de un cuadrado representado por un arreglo de ocupación espacial.

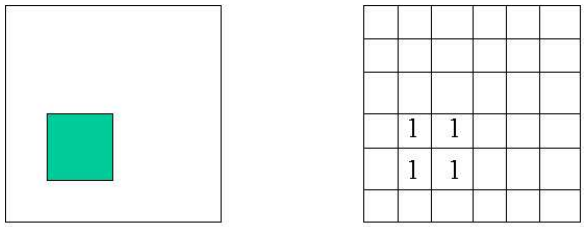


Figura 10.8: Arreglo de pertenencia espacial. En (a) se muestra una imagen de un cuadrado y en (b) el arreglo de ocupación espacial correspondiente

Aunque esta representación es fácil de implementar y facilita las operaciones lógicas entre regiones, es difícil hacer reconocimiento y es muy ineficiente en espacio.

Eje-Y

La representación "Eje-Y" onsiste en codificar la región en una serie de listas por renglón (elementos en Y), de forma que c/u representa las coordenadas en X donde se entra/sale de la región. Por ejemplo, la región en la figura 10.9 se codificaría de la siguiente forma:

$$(5, 15), (4, 16), (3, 16), ...$$

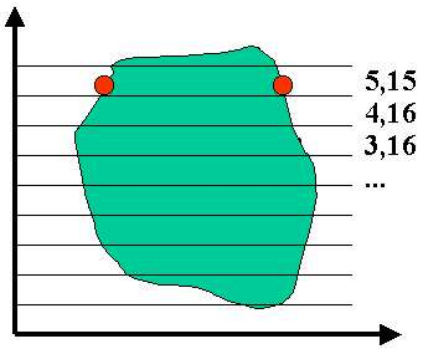


Figura 10.9: Codificación eje-Y. La región se codifica por la secuencia de coordenas en que cada renglón (líneas) entre y sale de la región.

Aunque esta representación es más eficiente en espacio, tampoco es muy conveniente para reconocimiento.

Árboles cuaternarios

Se utiliza una estructura en base a *quadtrees* para representar regiones. Para ello se considera la representación a varios niveles, donde cada cuadrante se marca como negro (pertenece a la región), blanco (no pertenece) y gris (pertenece parcialmente, ir al sig. nivel). La región consiste de la unión de los cuadrantes a varios niveles. La figura 10.10 muestra un ejemplo sencillo de una forma (avión) representada por un árbol.

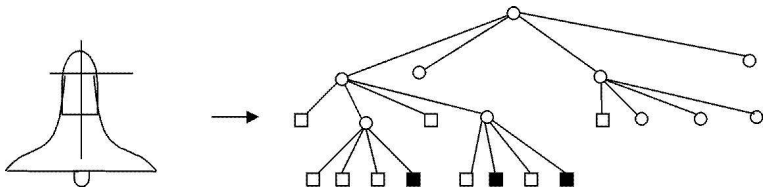


Figura 10.10: Una región en forma de avión (a) y el árbol cuaternario correspondiente (b). Los nodos representados como círculos corresponden a regiones ocupadas parcialmente, que tienen que descomponerse en el siguiente nivel; los cuadros rellenos representan regiones ocupadas por el avión y los cuadros blancos regiones que no pertenecen al avión.

Esta representación es bastante eficiente y facilita muchas operaciones como la obtención de área. Un problema es la restricción por la retícula predefinida.

Esqueletos

Si una región está formada de componentes delgados, una forma de representarla es mediante un "esqueleto" que corresponda a los ejes de cada componente. El esqueleto se puede obtener mediante un algoritmo de adelgasamiento que preserve conectividad, como lo es la *transformada de eje medio* (*medial axis transform*, MAT). Este consiste en encontrar todos los puntos del interior de la región para los cuales dos o más puntos del contorno están a la misma distancia (tambieén se conocen como diagramas de Voronoi). El conjunto de estos puntos forma el esqueleto. Otra forma de obtener esqueletos es mediante técnicas de *morfología matemática*. Ejemplos de algunas figuras sencillas y su esqueleto correspondiente se ilustran en la figura 10.11.

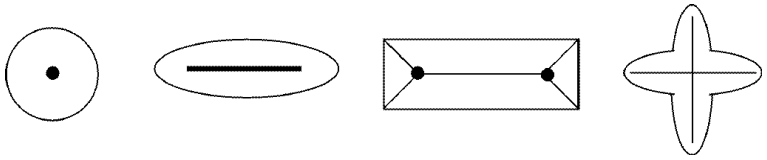


Figura 10.11: Ejemplos de esqueletos.

Un ejemplo de la aplicación de esqueletos es la representación de una mano (previamente segmentada), como podemos observar en la figura 10.12. En este ejemplo, el esqueleto es transformado en una serie de vectores que posteriormente son utilizados para reconocimiento de diferentes posiciones (ademanes) de la mano.

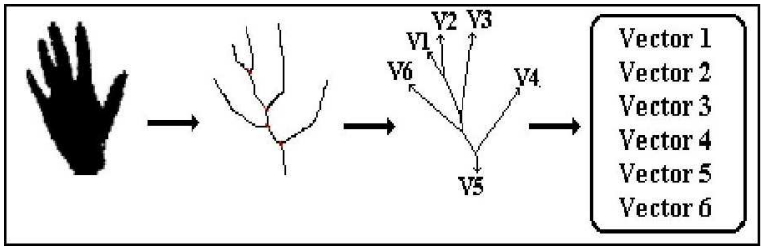


Figura 10.12: Esqueleto de una mano: (a) imagen binaria de la mano segmentada, (b) esqueleto, (c),(d) representación del esqueleto en base a vectores.

Un problema es que el esqueleto es muy sensible a ruido en el contorno, así como que se pueden producir esqueletos desconectados.

10.3.3 Descriptores globales

Los descriptores globales son propiedades simples de una forma bidimensional que ayudan a su descripción y reconocimiento en base a pocos atributos. Algunos descriptores globales usados comunmente son los siguientes:

- *Área*: Área total de la regió'n, se puede obtener facilmente de las representaciones de contornos y regiones.
- *Eccentricidad*: Medidida de la eccentricidad o "elongación" de la región. Una medida posible es la razón entre la cuerda máxima y mínima perpendiculares: $E = A/B$.
- *Número de Euler*: Descriptor de la topología de la región. NE = (número de regiones conectadas) - (número de hoyos).
- *Compactez*: Es la razón del perimetro respecto al área: $C = P^2/A$. Es mínima para un círculo.
- *Firmas*: Proyecciones de la región sobre diferentes ejes. Dada una imagen binaria, la firma horizontal se define como: $p(x) = \int_y f(x, y)$ y la vertical: $p(y) = \int_x f(x, y)$.
- *Números de forma*: Consiste en buscar un código único para curvas cerradas representadas por códigos de cadena. Para ello se selecciona la resolución adecuada y se orienta la figura de acuerdo a su diámetro mayor. Se obtiene el código de cadena y su derivada, que se normaliza rotándolo hasta encontrar el número mínimo.

Algunos de estos descriptores se ilustran en la figura 10.13

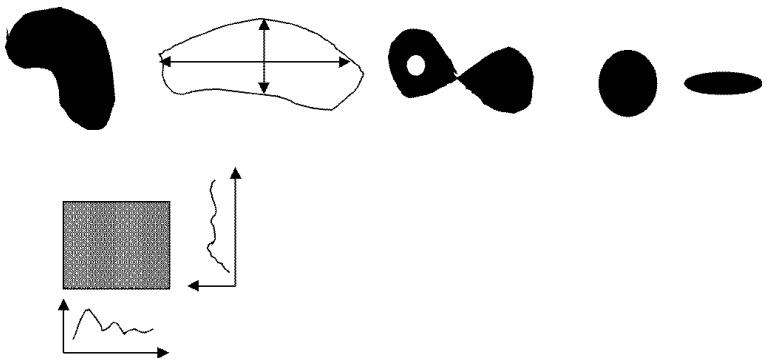


Figura 10.13: Descriptores globales: (a) área, (b) eccentricidad, (c) número de Euler, (d) compactez, (e) firmas.

10.4 Modelos en tres dimensiones

Los modelos en 2-D están restringidos a objetos bidimensionales o una sola vista de un objeto tridimensional. Para reconocer objetos en el mundo independientemente del punto de vista se requieren modelos tridimensionales.

Los modelos en tres dimensiones (3-D) consisten en una representación tridimensional de los objetos independientemente del punto de vista. Normalmente se asume que los objeto son sólidos y rígidos.

Al igual que en el caso de modelos en 2-D, los modelos en 3-D pueden ser básicamente de dos tipos: (i) modelos basados en una representación de la superficie del objeto, y (ii) modelos en base a una representación volumétrica del sólido. Existen diversos modelos de ambos tipos, en las siguientes secciones se describen 3 modelos representativos:

- poliedros planos,
- cilindros generalizados,
- geometría sólida constructiva.

10.4.1 Poliedros planos

Una representación muy útil, en base a superficies, consiste en aproximar el objeto mediante poliedros planos. Para esto se identifican las caras, aristas y vértice que se integran en una estructura mediante un grafo. En este grafo se tienen nodos y arcos que representan lo siguiente:

Nodos: representan los elementos de la representación: caras, vértices y aristas.

Arcos: representan relaciones espaciales entre los elementos. Existe una liga entre dos elementos si éstos son contiguos en la superficie.

Por ejemplo, la figura 10.14 muestra el modelo gráfico de un tetraedro.

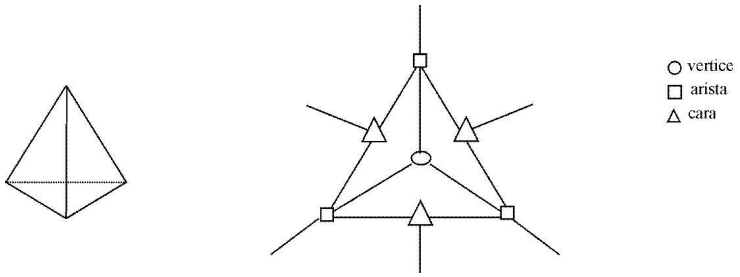


Figura 10.14: Representación de un tetraedro en base a poliedros planos. En (a) se muestra un tetraedro y en (b) parte del grafo correspondiente, indicando las caras, vértices y aristas, y las relaciones entre estos.

Esta es una representación muy poderosa y adecuada para reconocimiento, pudiendose utilizar información de orillas y regiones. Esta restringida a objetos que se puedan aproximar por poliedros planos, y aún en este caso la definición de las caras puede ser ambigua.

10.4.2 Cilindros generalizados

Los cilindros generalizados son una representación muy popular en visión, en la cual se define un eje mediante una función cualquiera y una superficie cerrada que "barre" este eje en forma perpendicular definiendo así un sólido. Matemáticamente se consistuye por dos funciones, una para el eje y otra para la superficie, la cual se especifica bajo un sistema de coordenadas local a cada punto del eje. La figura 10.15 ilustra un ejemplo de esta representación, en la cual se tiene como estructura base un círculo que cambia de tamaño a lo largo del eje de barrido.

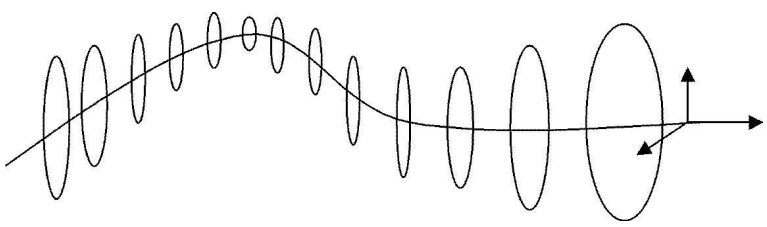


Figura 10.15: Ejemplo de una representación en base a cilindros generalizados.

Los cilindros generalizados se pueden extender considerando diferentes "cilindros" que se combinan en una estructura más compleja. Por ejemplo, para representar en una forma simplificada a una persona se pueden considerar 5 *cilindros*: uno para la cabeza y torso, dos para los brazos y dos para las piernas.

Esta representación es útil tanto para objetos naturales como artificiales.

10.4.3 Geometría sólida constructiva

La geometría sólida constructiva (CSG, pos sus iniciales en inglés) se basa en la composición de sólidos, generalmente simples, para formar objetos más complejos. Dichos sólidos primitivos se combinan mediante operaciones lógicas: unión, intersección y diferencia para formar objetos de mayor complejidad. En la figura 10.16 se muestra como se puede generar un modelo de un objeto mediante la composición de 3 formas básicas.

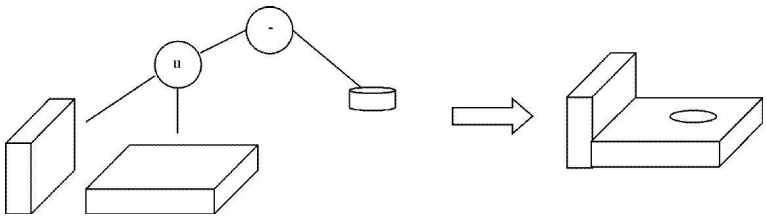


Figura 10.16: Geometría sólida constructiva. Mediante la unión (+) de dos prismas y la resta (-) de un cilindro (a), se contruye el objeto (b).

Este modelo es apropiado, principalmente, para objetos artificiales como partes para manufactura.

10.4.4 Propiedades de masa

Al igual que para 2-D, en 3-D podemos definir varias propiedades globales para objetos sólidos. Algunas de las propiedades comunmente utilizadas son las siguientes:

- Volumen: $V = \int_s du$
- Centroide: $C_x = \frac{\int_s x du}{V}$
- Momento de inercia: $I_{xx} = m \int_s (y^2 + z^2) du$
- Producto de inercia: $P_{xy} = m \int_s xy du$

10.5 Reconocimiento

El propósito final de visión es reconocimiento. Para ello es necesario realizar una interpretación de los datos de entrada. Esto se logra mediante el estableciemiento de una correspondencia (*match*) entre los atributos obtenidos de la(s) imagen(es) a través de los processo de visión de nivel bajo e intermedio y los modelos representados en la computadora. Este proceso se facilita si ambas representaciones son similares, por lo que los diferentes niveles de visión tienden a llegar a una descripción de la imagen que se asemeje a los modelos internos de los objetos del dominio de aplicación del sistema. Esto corresponde a la última parte de la estructura general de un sistema de visión basado en modelos como se representa en la figura 10.3.

Dependiendo del tipo de representación podemos distinguir 3 tipos de técnicas para correspondencia:

- Reconocimiento estadístico de patrones.
- Optimización paramétrica.
- Algoritmos basados en teoría de gráficas.

Las dos primeras se orientan a representaciones paramétricas (como aquellas basadas en descriptores globales) y la tercera se enfoca a estructuras relacionales (como los grafos en base a poliedros planos). En las siguientes secciones se presentan las 3 técnicas.

10.5.1 Reconocimiento estadístico de patrones

Si representamos los objetos mediante una serie de parámetros globales, como vimos para 2-D y 3-D, podemos aplicar técnicas de reconocimiento estadístico de patrones. Éstas consisten, básicamente, en buscar, dentro de un espacio paramétrico, la clase (modelo) más "cercana" a la descripción del objeto en la imagen. Si consideramos que se tienen, por ejemplo, dos parámetros y 3 tipos (clases) de objetos, el problema se puede visualizar como se representa en la figura 10.17. Cada punto en este representa un objeto. Los objetos similares se muestran con diferentes símbolos (x, +, *), los cuales, normalmente, están agrupados en el espacio paramétrico. Dado un objeto desconocido, el problema es encontrar a que grupo (clase) pertenece dados sus parámetros.

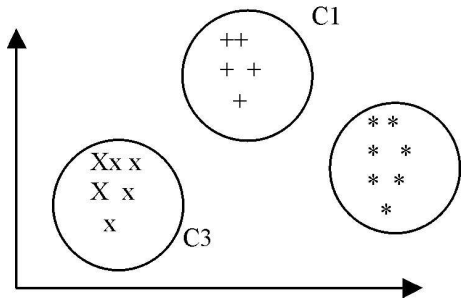


Figura 10.17: Espacio paramétrico con dos parámetros y tres clases. Cada círculo engloba objetos del mismo tipo.

Una técnica clásica para ésto se basa en teoría de probabilidad y se conoce como *teoría de decisiones bayesiana*. El modelo básico bajo este enfoque es el llamado *clasificador bayesiano*.

Clasificador bayesiano

El clasificado bayesiano se base en obtener la clase, C_i, más probable, dado un conjunto de atributos X. En su versión más sencilla se considera que los atributos son independientes dada la clase y se conoce como el *clasificador bayesiano simple* (*naive Bayes*).

Para obtener este modelo se requieren las siguientes probablilidades:

- $P(C_i)$, probabilidad *a priori* de cada clase
- $P(X_j|C_i)$, probabilidad condicional de cada atributo, X_j dada cada clase.

Éstas probabilidades se pueden estimar a partir de ejemplos conocidos de objetos (con sus respectivos atributos) de las diferentes clases.

La probabilidad condicional de que un patrón (observado) pertenezca a cierta clase, por el teorema de Bayes, es:

$$P(C_j|\vec{X}) = P(C_i)P(\vec{X}|C_i)/P(\vec{X}) \tag{10.8}$$

Donde $\vec{X}$ es el conjunto o vector N de atributos, $X_1, ..., X_N$. El denominador, $P(\vec{X})$, no depende de la clase por lo que es un valor constante. Entonces podemos escribr la ecuación anterior como:

$$P(C_j|\vec{X}) = KP(C_i)P(\vec{X}|C_i) \tag{10.9}$$

Donde "K" se puede considerar como una constante de normalización (hace que las probabilidades de las diferentes clases sumen uno). En el caso del clasificador bayesiano simple, el término $P(\vec{X}|C_i)$ se puede separar en el producto de las probabilidades individuales de cada atributo dada la clase:

$$P(C_j|\vec{X}) = KP(C_i)P(X_1|C_i)P(X_2|C_i)...P(X_N|C_i) \tag{10.10}$$

$$P(C_j|\vec{X}) = KP(C_i)\prod_{1}^{N} P(X_j|C_i) \tag{10.11}$$

Utilizando esta última expresión, se calcula la probabilidad posterior para todas las clases y tomamos la clase que de un valor mayor. En general, decidimos que cierta observación X pertence a la clase C_k de acuerdo a la regla de desición de Bayes:

$$g(C_k) > g(C_j), \forall j \neq k \tag{10.12}$$

Donde g puede ser directamente la probabilidad posterior u otra función de ésta. Por ejemplo, se puede tomar el logaritmo de las probabilidades o cualquier otra función monotónica. Para el caso de un atributo y dos clases, la regla de decisión bayesiana se ilustra en la figura 10.18. Si la probabilidad posterior dado un valor de X esta de lado izquierdo de la "línea de decisión" se selecciona $C1$, si no, $C2$.

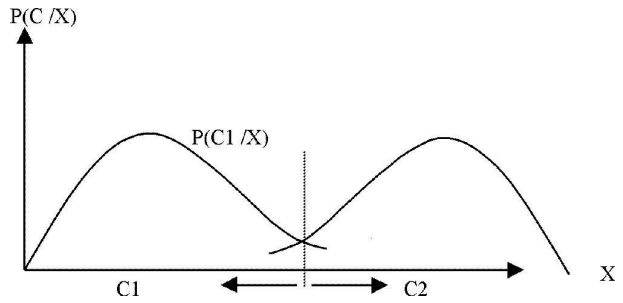


Figura 10.18: Descriminación basada en probabilidades.

Existen extensiones al clasificador bayesiano que consideran dependencias entre los atributos, así como otras técnicas de clasificación como las redes bayesianas, los árboles de decisión y las redes neuronales.

10.5.2 Optimización paramétrica

Las técnicas de optimización paramétrica se orientan a la correspondencia entre modelos paramétricos y representaciones de bajo nivel (por ejemplo, encontrar la correspondencia entre una serie

de orillas y una curva). Los modelos se describen por un vector de parámetros $\vec{a} = (a_1, a_2, ..., a_N)$. Se establece una función de mérito que mide que tan bien el modelo ($\vec{a}$) describe a los atributos de la imagen. De forma que el reconocimiento se plantea como un problema de optimización, donde se busca maximizar la siguiente función:

$$M(\vec{a}, f(x, y)) \tag{10.13}$$

Donde $f(x, y)$ son las atributos obtenidos de la imagen. Si M es una función "bien comportada", encontramos un máximo local cuando:

$$M_{a_j} = \frac{\partial M}{\partial a_j} = 0, j = 1, .., n \tag{10.14}$$

Para encontar este máximo se pueden usar diferentes tipos de técnicas:

- Técnicas analíticas - la función es simple y se puede encontrar el máximo analíticamente.
- Técnicas de gradiente (*hill climbing*) - se encuentra una solución aproximada que se va mejorando "moviéndose" en la dirección del gradiente.
- Perturbación de los coeficientes - si la derivada es difícil de obtener, se modifican ligeramente los coeficientes (partiendo de una solución inicial), en forma aleatoria o estructurada, y se mantienen si mejoran M.

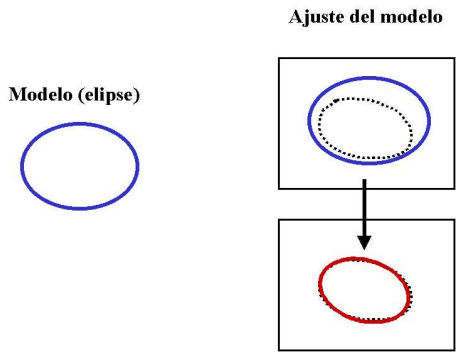


Figura 10.19: Ejemplo de optimización paramétrica. El modelo de la elipse (izquierda) se ajusta a las características -orillas- obtenidas en la imagen (derecha).

En la figura 10.19 se ilustra el proceso en forma gráfica. En este caso el modelo es una elipse, la cual se "ajusta" al contorno obtenido de la imagen mediante la modificación de sus parámetros.

10.5.3 Algoritmos basados en teoría de grafos

Los algoritmos basados en teoría de grafos, en particular el *isomorfismo* de grafos, se aplican cuando tenemos una representación relacional, tanto de los modelos internos como de la descripción de la imagen. Se considera que ambas están representadas en forma de un grafo (nodos y relaciones), como en el caso de la representación de poliedros. Entonces el problema es encontrar la correspondencia entre dichos grafos.

Desde el punto de vista de teoría de grafos esta correspondencia se refiere al problema de *isomorfismo entre grafos*. En su forma pura consiste en encontrar una relación 1:1 entre arcos y nodos de ambos grafos, considerando que no están etiquetados. Un ejemplo se ilustra en la figura 10.20, donde se muestran dos grafos isomórficos. En la práctica, se consideran correspondencias

parciales, y también que los nodos y arcos tienen etiquetas de forma que se pueden aparear con sólo ciertos otros (con la misma etiqueta).

Existen 3 tipos de isomorfismos entre grafos:

- Isomorfismo de grafos: correspondencia 1:1 entre dos grafos, $G1$ y $G2$.
- Isomorfismo de subgrafos: correspondencia entre una grafo $G1$ y los subgrafos de $G2$.
- Doble ismorfismo de subgrafos: encontar todos los isomorfismos entre subgrafos de $G1$ y subgrafos de $G2$.

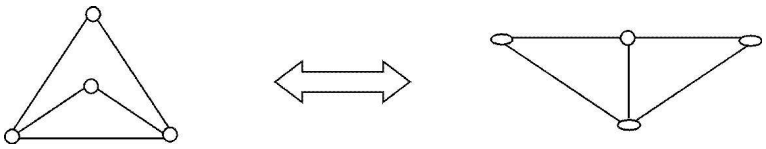


Figura 10.20: Isomorfismo de grafos. Los grafos de la izquierda y derecha son isomorfos.

El segundo caso es más complejo que el primero, y aunque el tercero es aparentemente más complejo, se puede demostrar que ambos (isomorfismo de subgrafos y doble isomorfismo) son equivalentes. El isomorfismo de subgrafos es, en el peor caso, un problema NP-completo; pero existen algoritmos que dan en el caso promedio tiempos proporcionales a N^3 y N^2 (N = número de nodos). Existen diversas técnicas para resolver los problemas de isomrofismo entre grafos y subgrafos, nosotros consideraremos 2 alternativas que se describen a continuación.

Búsqueda con *backtracking*

La técnica de búsqueda con *backtracking* consiste en hacer una búsqueda exhaustiva por profundidad en un árbol de soluciones parciales. Para ello se considera el problema de isomorfismo de subgrafos entre $G1$ y $G2$. El procedimiento es el siguiente:

1. Se contruye el árbol con un nodo incicial (vacío).
2. Se inicia con un nodo de $G1$ y todas las posibles correspondencias con $G2$ (primer nivel).
3. Se buscan todas los nodos conectados al nodo incial en $G1$ y su correspondencias en $G2$ (segundo nivel), de forma que haya correspondencia entre arcos.
4. Se repite (3) hasta que ya no existan correspondencias o se hayan considerado todos los nodos (niveles 3 al N).

De esta forma se va creando un árbol cuyas trayectorias hasta el nivel n corresponden a los isomorfismos de $G1$ y $G2$. La aplicación a un ejemplo sencillo del método se muestra en le figura 10.21. En este ejemplo, se tiene 3 tipos de nodos (A, B, C), de forma que deben de corresponder nodos del mismo tipo.

Búsqueda de cliques

Un *clique* (conglomerado) es un conjunto de nodos (N), en un grafo, los cuales están todos conectados entre sí, formando una subgrafo totalmente conectada de tamaño N (existe un arco entre cada nodo y los demás). Para encontrar doble isomorfismo se construye una grafo asociativo G, entre los 2 grafos, $G1$, $G2$, y se encuentran los cliques en G. La búsqueda de cliques es similar en

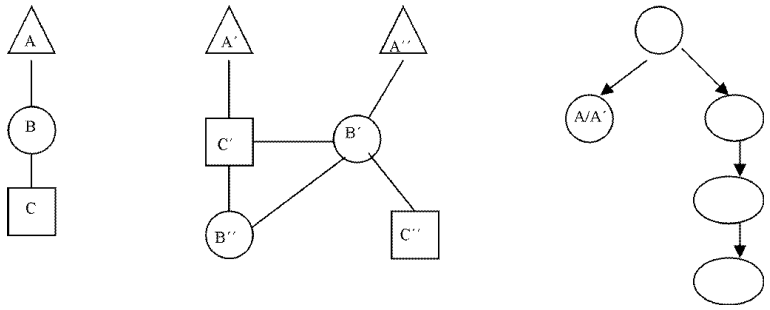


Figura 10.21: Ejemplo de isomorfismo por búsqueda. (a) Grafo $G1$. (b) Grafo $G2$. (c) Árbol de soluciones: se encuentran dos isomorfismos, uno de un solo nodo (AA'), y otro de 3 nodos, el grafo $G1$ y un subgrafo de $G2$.

complejidad al problema de isomorfismo de subgrafos, por lo cual isomorfismo de subgrafos sencillo y doble son equivalentes.

El grafo asociativo G se construye de la siguiente forma:

1. Para cada par de nodos compatibles de $G1$ y $G2$ contruir un nodo V_i en G.
2. Construir una liga entre nodos de G, V_i, V_j, si las ligas entre los nodos correspondientes en los grafos originales son compatibles.
3. Se buscan los cliques en el grafo asociativa G, los cuales indican correspondencias parciales.
4. El clique de mayor tamaño indica el mejor *match*.

Un ejemplo de isomorfismo por búsqueda de cliques se presenta en la figura 10.22. Se tiene dos grafos, uno completo que corresponde a la imagen y otro parcial que corresponde a las características obtenidas de la imagen. Se muestra el grafo asociativo de ambos, donde cada clique se ilustra con diferentes tipos de nodos. En este caso se tienen 3 cliques de tamaño 4.

10.6 Ejemplos de aplicaciones

Algunos ejemplos de aplicaciones de sistemas basados en modelos se describen a continuación.

Shirai [107] presenta una aplicación para reconocer objetos en un escritorio (modelo en 2D). Busca reconocer objetos típicos en un escritorio (teléfono, lámpara, etc.), representados por sus características principales en líneas y elipses. Usa un algoritmo iterativo, localizando primero el atributo principal (rueda del teléfono) y regresando a nivel-bajo para encontrar otros atributos.

Ballard [2] utiliza una representación basada en cilindros generalizados para reconocimiento de objetos curvos en 3-D. El reconocimiento se basa en correspondencia entre grafos utilizando semántica para simplificar la b'usqueda. Para ello se utiliza un índice a los modelos en base a sus características principales. Se ha probado con modelos de 5 objetos (muñeca, caballo, etc.).

El programa *ACRONYM* [9] utiliza modelos parametrizados en base a cilindros generalizados para reconocimiento de objetos en 3-D. El reconocimiento se basa en predecir las imágenes en 2-D de los modelos y encontrar su correspondencia con las características extraidas de la imagen. Para ello utiliza un sistema de proyección algebraica y un sistema de manipulación de restricciones con heurísticas. Se ha aplicado a imágenes de aviones.

Los sistemas basados en modelos se aplican, principalmente, para reconocimiento de objetos artificiales, como en sistemas industriales. Tienen 3 restricciones:

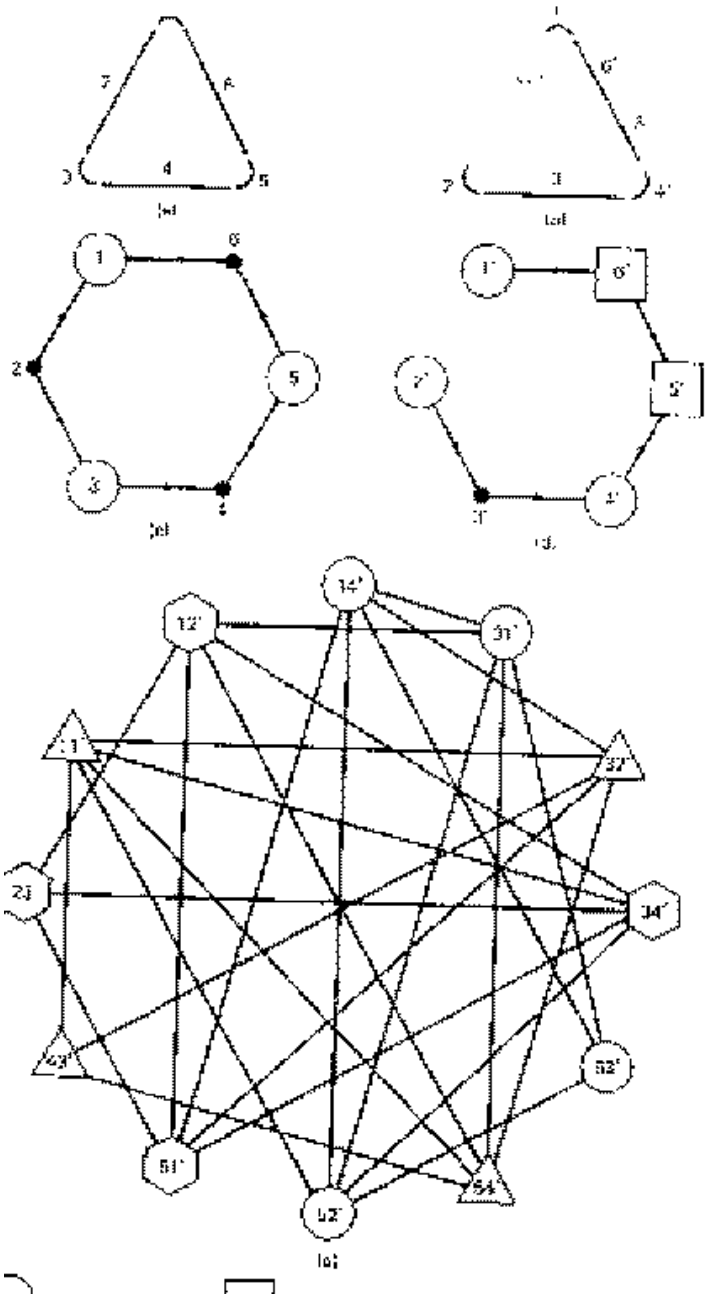


Figura 10.22: Grafo asociativo y cliques. (a) Modelo, (b) características de la imagen, (c) grafo del modelo, (d) grafo de la imagen, (e) grafo asociativo con cliques indicados mediante la forma de los nodos.

1. Consideran modelos simples, geométricos y con pocos parámetros.
2. Contienen pocos objetos en su dominio (complejidad computacional).
3. Asumen que la extracción de características es robusta y confiable.

10.7 Referencias

Las restricciones geométricas han sido utilizadas desde los primeros sistemas de visión. Uno de los primeros trabajos en utilizar modelos 2D de proyecciones en 3D es el de Roberts [98] en

donde se identificaban cubos, hexágonos y pirámides en base a sus orillas. El *match* se realizaba contra la proyección de los modelos 3D en memoria. Parte de la motivación de utilizar modelos tridimensionales se basa en estudios psicológicos. Piaget y Inhelder [91] estudiaron el desarrollo visual en niños, encontrando que a partir de siete u ocho años de edad pueden decidir si dos vistas corresponden o no a un mismo objeto. Esto permite concluir que a partir de esa edad pueden anticipar los efectos de las rotaciones rígidas. Desde el punto de vista de visión esto se podría interpretar como construir un modelo tridimensional a partir de una vista y poder rotarlo (reconstruyendo la información ocluída. Después se necesita calcular la proyección 2D de este modelo interno y hacer el "match" contra la nueva vista. Las transformaciones para reconocimiento de patrones, tales como rotación o escalamiento, han sido estudiadas por S. Ullman [42, 129].

A. Guzmán [29] estendió el trabajo de Roberts y consideró no utilizar un modelo 3D sino la información de las junturas. El dibujo de entrada era procesado para identificar objetos polihedricos en base a las junturas válidas de los objetos prismáticos convexos. Huffman [36] y Clowes [16] generalizó la idea de A. Guzmán para etiquetar las orillas como concavas, convexas u ocluídas pero basó su trabajo en un máximo de tres junturas. Por ejemplo, una pirámide de base cuadrada quedaría excluída. El etiquetado producía una gran cantidad de modelos igualmente válidos, pero dejaba de utilizar la construcción heurística de Guzmán (a costa de una enumeración exhaustiva y de generar múltiples soluciones todas válidas). D. Waltz [133] extendió el trabajo de Huffman introduciendo más tipos de juntura y orillas. En vez de hacer enumeración exhaustiva propagaba las posibles junturas, basandose en un "diccionario de junturas", y eliminando las inconsistentes desde un inicio. Este *filtrado Waltz* permitía remover las etiquetas imposibles y generaba, en el peor de los casos, un reducido número de objetos válidos. Este tipo de filtrado es considerado como pionero en la aplicación de técnicas de relajación en visión. Aun más, T. Kanade [50] extendió el "block-world", en el cual se basaban los métodos anteriores, para introducir el "origami-world". Esta representación permitía no solo los polihedros sólidos sino cualquier objeto que pueda ser descompuesto por superficies planas. Desafortunadamente, su *surface connection graph* permitía, aún despúes de eliminar las caras inconsistentes (dos superficies con dos orientaciones diferentes), múltiples interpretaciones. Su último trabajo sobre el tema [51] concluye que es necesario tomar en cuenta las regularidades (i.e. líneas paralelas), la misma conclusión a la que llegan que trabajan en agrupamiento perceptual [71, 115].

Para mayores referencias y técnicas, se recomiendan los siguientes trabajos que hacen una revisión bibliográfica del área [6, 131, 70].

10.8 Problemas

1. Encuentra y demuestra matemáticamente una fórmula para calcular el área de un polígono a partir de su representación en polilíneas.

2. Plantea modelos en base a poliedros, cilindros generalizados y CSG para una tuerca hexagonal, un árbol y una persona.

3. Resuelve el problema de isomorfismo de subgrafos de la figura 10.22 mediante la técnica de búsqueda con *backtracking*.

4. Dada las siguentes estructuras relacionales:

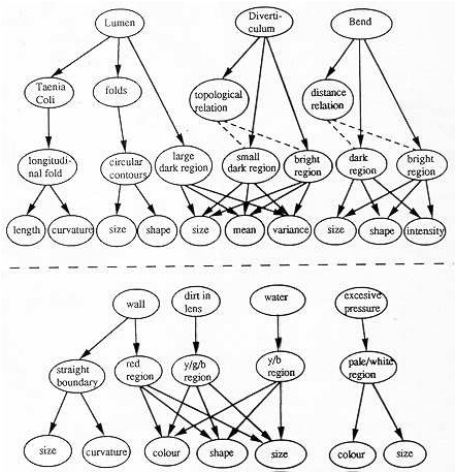
   A: a,b,c,d,e,f. Relaciones: R1(a), R1(c), R1(e), R2(b), R2(d), R2(f), R3(a,b), R3(b,c), R3(c,d), R3(d,e), R3(e,f), R3(f.a)

   B: u,v,w,x,y,z. Relaciones: R1(u), R1(v), R1(x), R2(w), R4(y), R4(z), R3(v,w), R3(w,x), R3(x,y), R3(y,z), R3(z,u)

   Obten: (a) La gr'afica correspondiente a cada una, etiquetando nodos y arcos, (b) La gr'afica asociativa entre A y B, (c) Los cliques m'aximos en dicha gr'afica asociativa.

5. Dadas imágenes de polígonos regulares, como tríangulos, rectángulos, pentágonos, plantea un método para reconocer el tipo (clase) de polígono independiente de su tamaño y posición en la

imagen. (a) Utiliza un método basado en reconocimiento estadístico de patrones, indicando los atributos a utilizarse (b) Utiliza un método basado en teoría de grafos. Para (a) y (b) haz un diagrama de bloques detallado del proceso indicando las operaciones en cada bloque.



Capítulo 11

Visión Basada en Conocimiento

11.1 Introducción

Los sistemas de visión basados en conocimiento utilizan modelos proposicionales para su representación, a diferencia de los basados en modelos que utilizan representaciones analógicas. Tienen una colección de proposiciones que representan conocimiento sobre los objetos y sus relaciones. El reconocimiento se realiza mediante un proceso de inferencia. A partir de los datos de la imagen y el conocimiento del dominio se infiere la identidad de los objetos en la imagen. En la figura 11.1 se ilustra la arquitectura general de un sistema de visión basado en conocimiento.

Un sistema de visión basado en conocimiento consta de 3 procesos principales:

1. Extracción de características - obtener los atributos importantes de la imagen(es) mediante visión de nivel bajo/medio e integrarlos en una *imagen simbólica*.
2. Representación del conocimiento - contrucción del conocimiento sobre el dominio. Esto se hace previamente, guardándose en la *base de conocimientos*.
3. Inferencia - proceso de deducir de la imagen simbólica y la base de conocimiento la identidad y localización de los objetos de interés.

La visión basada en conocimiento se deriva de lo que se conoce como sistemas basados en conocimiento o sistemas expertos. Éstos son sistemas que resuelven problemas mediante procesos

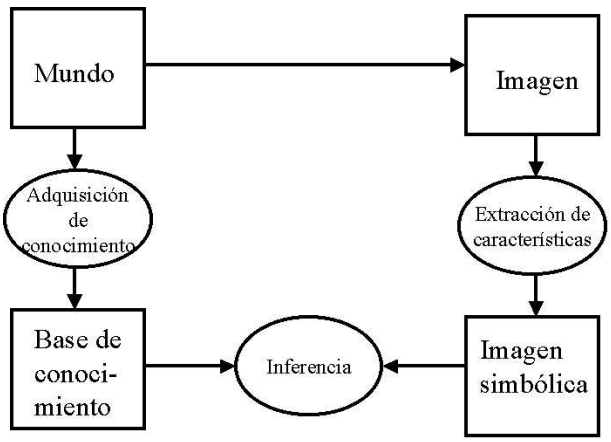


Figura 11.1: Sistema de visión basado en conocimiento.

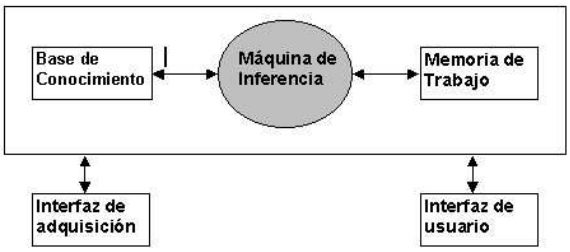


Figura 11.2: Arquitectura de un sistema basado en conocimiento.

de razonamiento utilizando una representación simbólica del conocimiento humano. El aspecto fundamental de este tipo de sistemas es como se representa el conocimiento, lo que influye también en la forma de razonamiento. Algunas de las principales representaciones utilizadas en visión son las siguientes:

- reglas de producción,
- redes semánticas,
- prototipos (*frames*),
- redes probabilísticas o redes bayesianas.

Otro tipo de sistemas utilizan representaciones basadas en modelos biológicos aproximados del cerebro humano. Éstos se conocen como redes neuronales y han sido también aplicados en visión. A continuación describiremos cada una de estas representaciones y su aplicación en visión. Antes, veremos una descripción general de lo que es un sistema basado en conocimiento o sistema experto.

11.2 Sistemas basados en conocimiento

Los sistemas basados en conocimiento o sistemas expertos tienen conocimiento de un dominio particular, el cual utilizan mediante un proceso de inferencia para resolver problemas específicos. Dicho conocimiento se encuentra generalmente expresado en forma simbólica, utilizando un proceso deductivo para a partir de los datos y el conocimiento llegar a ciertas conclusiones. Tienen 3 partes principales:

- Base de conocimiento - almacena el conocimiento del dominio.
- Memoria de trabajo - almacena los datos y conclusiones a que llega el sistema.
- Máquina de inferencia - realiza el proceso de razonamiento, aplicando el conocimiento a los elementos en la memoria de trabajo.

En la figura 11.2 se muestra una arquitectura general de un sistema basado en conocimiento.

Los sistemas expertos representan en una forma explícita el conocimiento, generalmente sobre un dominio específico. El conocimiento se puede expresar de diferentes formas, entre las más comunes se encuentran:

- lógica proposicional,
- lógica de predicados,

- reglas de producción,
- redes semánticas,
- *frames* (prototipos o marcos).

La capacidad de representar diferentes tipos de conocimiento (expresividad) y la velocidad para poder hacer inferencias (eficiencia) varía para las diferentes representaciones. Algunas, como la lógica proposicional, son poco expresivas y muy eficientes; otras, como la lógica de predicados, son muy expresivas pero ineficientes; mientras otras representaciones buscan un compromiso entre ambos aspectos. Nos enfocaremos a éstas representaciones: reglas de producción, redes semánticas y *frames*, en su aplicación a visión de alto nivel.

En visión generalmente existe incertidumbre, debido a varios factores: ruido, proceso de adquisición y digitalización, errores en el procesamiento de bajo nivel, conocimiento incompleto, oclusiones, etc. Las representaciones anteriores, en general, no consideran en forma explícita y adecuada la incertidumbre. Existen otras formas alternativas que toma en cuenta le incertidumbre. Entre estas están las redes bayesianas y la lógica difusa. Veremos más adelante la aplicación de redes bayesianas a visión.

La forma de representación es fundamental para el rendimiento de un sistema basado en conocimiento. No existe una mejor representación para todos los problemas, ésta depende del dominio de aplicación. Para visión se han establecido ciertos criterios que debe satisfacer una buena represntación, los cuales se detallan a continuación.

11.3 Criterios de representación

Para comparar las diferentes formas de representación, se pueden definir una serie de "criterios de adecuación" para su aplicación en visión. Dichos criterios se dividen en dos tipos: descriptivos y procedurales.

Los criterios descriptivos nos dicen que tan adecuada es la representación para describir o representar el mundo. Los principales criteriors procedurales son:

1. Capacidad. Representación de diferentes situaciones o configuraciones.
2. Primitivas. Objetos primitivos del dominio, sus atributos y relaciones.
3. Composición. Representación de objetos estructurados.
4. Especialización. Generación de refinamientos de clases de objetos.
5. Submundos. Capacidad de mantener la distinción entre diferentes "submundos" (por ejemplo, entre 2-D y 3-D).
6. Proyección. Relación de los objetos en el mundo y en la imagen.
7. Clases equivalentes. Capacidad de representar escenas equivalentes.
8. Detalle. Representación a diferentes niveles de detalle o escala.
9. Estabilidad. Pequeños cambios en el mundo causan cambios pequeños en la representación.
10. Invariante. La representación debe ser invariante a transformaciones del mundo.
11. Correcta. Debe haber una relación funcional de situaciones a su representación. En particular, una sitaución anómala no debe tener una representación coherente; y una situación ambigua, debe tener dos o más posibles representaciones.

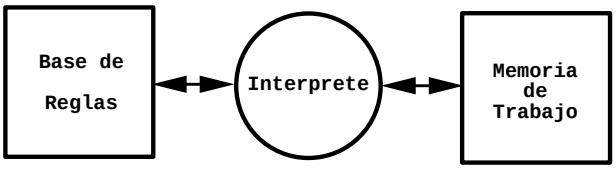


Figura 11.3: Sistema de producción.

Los criterios procedurales nos dicen que tan adecuada es la representación para el uso y adquisición del conocimiento, y son los siguientes:

1. Correctez. El sistema produce sólo interpretaciones permitidas por la representación.
2. Completez. El sistema produce todas las interpretaciones permitidas por la representación.
3. Flexibilidad. Utilización de todas las fuentes de información, en diferentes sentidos. Debe permitir el flujo de control de imagen a escena (análisis) o de escena a imagen (síntesis).
4. Adquisición. Facilidad de adquisición o aprendizaje de conocimiento de la representación.
5. Eficiencia. Rendimiento en tiempo y espacio, peor caso y promedio, de la representación y técnicas de inferencia asociadas.

Los criterior anteriores sirven de base para evaluar y comparar diferentes representaciones para visión, además de proveer una guía para desarrollar nuevas representaciones.

11.4 Reglas de producción

En los sistemas de producción el conocimiento se representa por un conjunto de reglas *condición - acción* de la forma:

SI $P1 \wedge P2 \wedge ... \wedge Pn \rightarrow Q1 \wedge Q2 \wedge ... \wedge Qm$

Donde cada premisa / conclusión es una tripleta *objeto-atributo-valor*. Por ejemplo, la siguiente es una regla sencilla para identificar un tipo de objeto (lumen) en imágenes endoscópicas:

SI (región.tamaño $>$ 16) & (región.media $=$ 20) ENTONCES (región.tipo $=$ lumen)

Un sistema de reglas normalmente tiene un número considerable (cientos o miles) de reglas, que en conjunto representan el conocimiento de un cierto dominio para una cierta tarea. Las reglas se almacenan en la memoria de producción de donde son ejecutadas por el interprete de acuerdo a un ciclo iterativo que consiste de 3 partes:

1. *Matching* - buscar las reglas cuyas conclusiones se encuentren en la memoria de trabajo.
2. Resolución de conflicto - escoger una de dichas reglas (criterios de especificidad, reciente, etc.).
3. Ejecución - aplicar la regla seleccionada modificando al memoria de trabajo.

Las estructura de un sistema de reglas de producción se muestra en la figura 11.3. Al igual que un sistema basado en conocimiento, consta de 3 partes principales: base de reglas, interprete y memoria de trabajo. La base de reglas almacena el conjunto de reglas del sistema. El interprete ejecuta el ciclo de selección y aplicación de las reglas. Los datos de entrada, conclusiones generadas por las reglas y datos de salida se almacenan en la memoria de trabajo.

A continuación se presenta la aplicación de reglas de producción en visión.

11.4.1 SPAM

Un ejemplo típico de la aplicación de reglas en reconocimiento de objetos en imágenes es el sistema SPAM. SPAM [Mckeown 85] es un sistema para la interpretación de imágenes aéreas de aeropuertos. Descompone la representación de un aeropuerto en 4 niveles:

1. Regiones - segmentación de los niveles bajos.
2. Fragmentos - posibles interpretaciones para una región.
3. Áreas funcionales - composición de varias regiones que representan un área funcional del aeropuerto.
4. Modelos - conjunto de áreas funcionales que representan un aeropuerto.

Tiene una serie de reglas para segmentación e interpretación que se dividen en 7 grupos:

- Inicialización.
- Interpretación inicial de regiones.
- Procesamiento de imágenes y agrupamiento de regiones.
- Consistencia de fragmentos.
- Agrupamiento y consistencia de áreas funcionales.
- Generación de metas (conocimiento general de aeropuertos).
- Agrupamiento de áreas funcionales en modelos.

Las reglas tienen "valores de confidencia" para la selección entre varias posibles hipótesis.

SPAM hace una interepretación de las imágenes de aeropuertos en base a las reglas, utilizando un enfoque de abajo hacia arriba. Primero identifica las regiones, después las agrupa en regiones, identifica áreas funcionales y finalmente el aeropuerto.

11.5 Redes semánticas

En las redes semánticas, el conocimiento se representa mediante una red, donde los nodos representan conceptos y las ligas representan diferentes tipos de relaciones entre ellos. Dicha red forma una jerarquía de conceptos relacionados, donde cada uno se representa en términos de otros. Existen diferentes tipos de ligas como operadores lógicos y relaciones de pertenecia. Un tipo de liga importante es "ISA", que denota que un un concepto o clase es una subclase de otra, permitiendo así la herencia de propiedades entre conceptos. Un ejemplo sencillo de una red semántica se ilustra en la figura 11.4.

Una red semántica se puede ver como una representación analógica o proposicional. En el primer caso el reconocimeinto se base en un proceso de correspondencia como en los sistemas basados en modelos, en le segundo caso se aplican reglas de inferencia operando en la estructura de la red. A continuación se ilustra como utilizar este enfoque en visión.

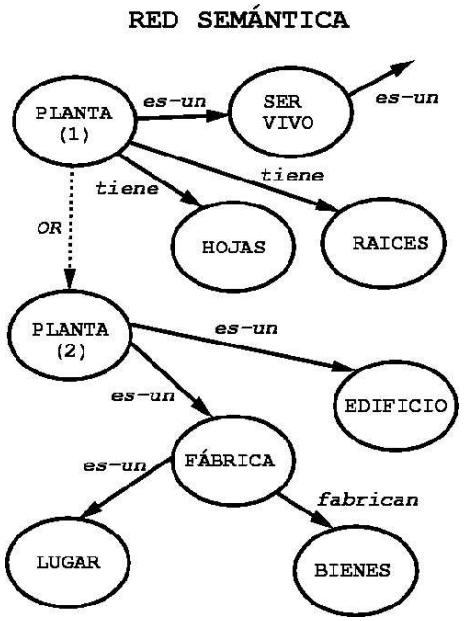


Figura 11.4: Ejemplo de una red semántica. Esta red representa el concepto de "planta", tanto desde el punto de vista de planta como ser vivo como de planta industrial.

11.5.1 Análisis dirigido por conocimiento

Ballard propone un sistema basado en redes semánticas para reconocimiento de objetos complejos en imágenes. Este sistema se divide en 3 estructuras principales: imagen, mapa y modelo. En la estructura de imagen se guarda la imagen original y características obtenidas de visión de bajo nivel. Los modelos son redes semánticas que representan objetos prototípicos del dominio de interés. El mapa es otra red semántica que se genera en el momento de la interpretación, y que relaciona la información en la imagen con la del modelo. Cada nodo del mapa se liga al nodo correspondiente del modelo y la estructura de la imagen. La construcción del mapa se realiza por una colección especializada de procedimientos de mapeo, que son particulares para cada dominio. El reconocimiento se logra mediante una correspondencia correcta en el mapa.

Este sistema ha sido aplicado al reconocimiento de radigrafías e imágenes aéreas.

11.6 Prototipos

Un prototipo o marco (*frame*) se define como "una estructura para la representación de una situación estereotípica". Un marco se puede ver como un especie de *record* que tiene una serie de registros que se agrupan en dos niveles: alto y bajo. Los registros de nivel alto son fijos y corresponden a características siempre ciertas. Los registros de bajo nivel son llamados terminales y se les asignan valores para cada caso. Pueden existir una serie de condiciones que deben satisfacer dichos registros terminales, y también pueden tener "defaults". Una colección de marcos se constituyen en un sistema de marcos, los cuales se ligan generalmente por relaciones de clase/subclase (ISA) en forma análoga a las redes semánticas. Un ejemplo de un sistema de *frames* se ilustra en la figura 11.5. Los marcos en esta sistema o jerarquía, *heredan* los atributos de sus "ancesestros", es decir, de los *frames* que están por arriba en dicha jerarquía.

El reconocimiento se basa en encontrar el marco "más cercano" a un situación determinada (imagen), asignándole valores a los nodos terminales. En visión, un marco representa una clase de objetos mediante un prototipo adaptado de una instancia particular. Diferentes marcos pueden representar un objeto desde diferentes puntos de vista. La aplicación de *frames* en visión se presenta

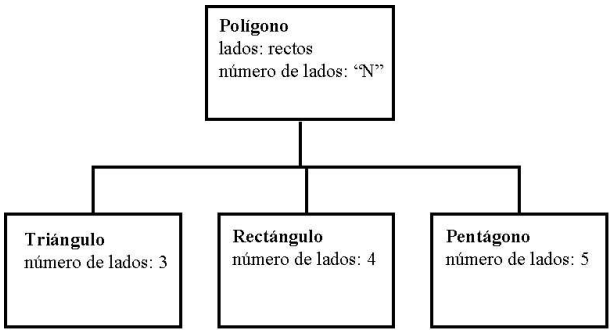


Figura 11.5: Ejemplo de un sistema de *frames*. Se tiene un marco general, "polígono", y tres marcos que son casos particulares: triángulo, rectángulo y pentágono. Estos tres marcos, heredan el atributo *lados* del marco superior.

a continuación.

11.6.1 Prototipos en visión

VISIONS es un sistema cuya representación se basa en marcos o *esquemas* que representan el prototipo de una escena. Éstos se agrupan en una jerarquía, desde marcos en el nivel superior que representan escenas completas, hasta el nivel inferior que corresponden a características de la imagen. Los marcos a diferentes niveles se ligan dentro de la jerarquía. Se tiene 7 niveles: escenas, objetos, volumenes, superficies, regiones, segmentos y vértices. Se tiene una memoria de largo plazo (LTM) donde se encuentra el conocimiento general del dominio, y una memoria de corto plazo (STM) que representa la interpretación de la escena bajo análisis. El esquema general de VISIONS se muestra en la figura 11.6.

El proceso de interpretación consiste en construir el esquema en STM usando el conocimiendo en la LTM y las características obtenidas de la imagen. Esto se logra mediante una serie de procedimientos llamados *fuentes de conocimiento*. Estos tienen conocimiento visual de diferentes propiedades (color, textura, etc.) y usan la información en LTM para construir hipótesis en STM. Para lograr dicha correspondencia, utlizan medidas de la contribución de la característica i al objeto j (C_{ij}) y su capacidad descriminatoria (W_{ij}). Se combinan todas las características de una región de la imagen para obtener la confidencia de que corresponda a la clase objeto:

$$confidencia = \sum_j W_{ij} C_{ij} \tag{11.1}$$

VISIONS se ha aplicado a la interpretación de escenas naturales (casas). Un ejemplo de esta aplicación es el que se ilustra en la figura 11.6.

11.7 Redes probabilísticas

Un problema en las representaciones anteriores (reglas, redes semánticas, rpototipos) al aplicarse a visión es el manejo de incertidumbre. Existe incertidumbre en visión por diversas causas:

- Ruido y distorsión en el proceso de adquisición y digitalización.
- Información incompleta e inconfiable de los procesos de nivel bajo.
- Dificultades propias de la imagen, como oclusiones, sombras y especularidades.

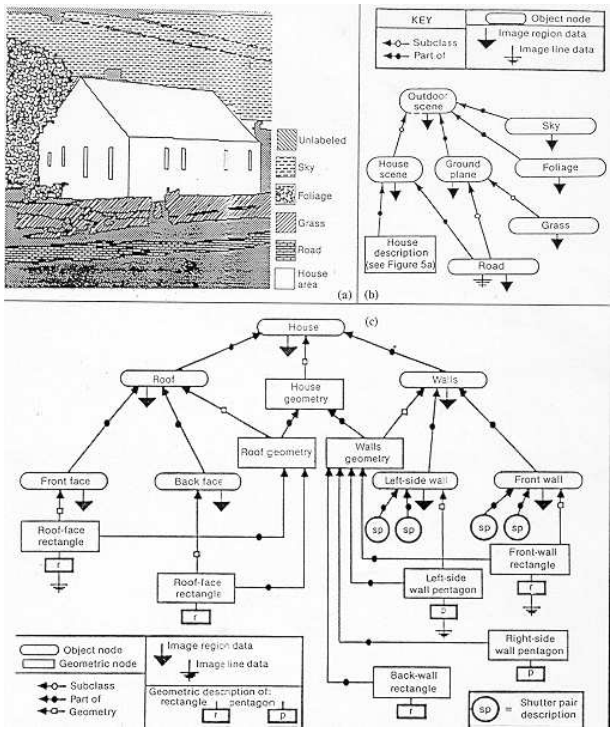


Figura 11.6: VISIONS: (a) imagen de un paisaje, (b) esquema general (LTM) de una escena de exteriores, (c) esquema particular (STM) contruido para la imagen a partir del esquema general.

Se han agregado formas de manejo de incertidumbre a las representaciones anteriores, pero éstas son usualmente *ad-hoc*, por lo que es difícil generalizarlas a otros dominios. Una representación que maneja incertidumbre en forma adecuada son las redes probabilísticas, también conocidas como redes bayesianas o causales.

Una red probabilística es una gráfica acíclica dirigida (DAG), donde cada nodo representa una variable y las ligas representan relaciones probabilísticas entre ellas, cuantificadas mediante probabilidades condicionales. Dichas ligas representan, normalmente, relaciones causales, de forma que una liga de A hacia B indica que A *causa* B. Un ejemplo de una red probabilística se presenta en la figura 11.7. Este ejemplo representa, en forma muy simplificada, una RP para distinguir entre una moneda y una pluma en una imagen. La moneda puede "producir" una imagen de un círculo. La pluma, dependiendo del punto de vista, puede ser un rectángulo y, con baja probabilidad, también un círculo.

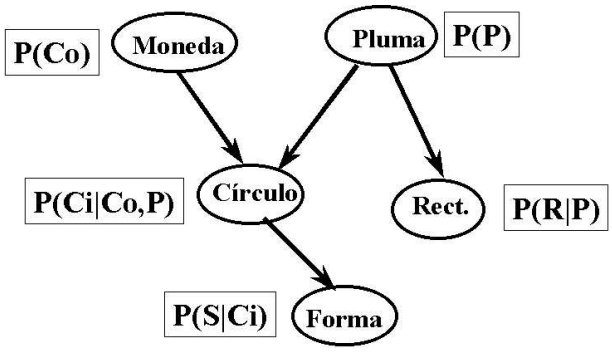


Figura 11.7: Ejemplo de una red probabilística. Cada variable (nodo) en la red tiene asociada una matriz de probabilidad condicional dados sus padres.

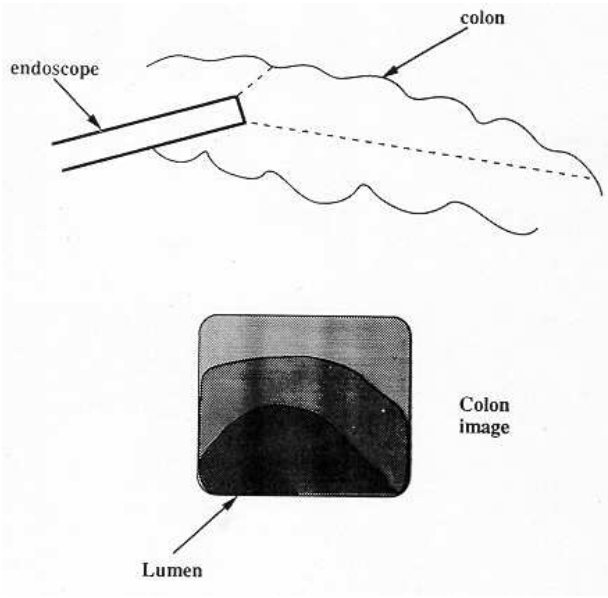


Figura 11.8: Endoscopía. En (a) se ilustra en forma esquemática el endoscopio dentro del tubo digestivo. Una imagen simplificacada que obtiene el endoscopio se muestra en (b), indicando el centro o *lumen* del colon.

Las probabilidades se pueden obtener subjetivamente (de un experto) o en forma objetiva de estadísticas del dominio. Dada una red probabilística se pueden realizar inferencias, obteniéndo lo probabilidad posterior de ciertas variables desconocidas a partir de otras conocidas mediante un mecanismo de razonamiento probabilístico. Este se basa el teorema de Bayes y consiste en propagar los efectos de las variables instanciadas (conocidad) a través de la red, para obtener las probabilidades posteriores de las variables desconocidas.

11.7.1 Redes probabilísticas en visión

Para visión, podemos considerar una red probabilística jerárquica organizada en una serie de niveles. Los niveles inferiores corresponden a las características de la imagen, y los niveles superiores a los objetos de interés. Los nivels intermedios corresponden a regiones, partes de objetos, etc.

En el proceso de reconocimiento, se instancian los nodos inferiores, propagándose su efecto hacia arriba hasta llegar al nivel superior. De esta forma se obtiene la probabilidad posterior para cada objeto, seleccionándose como interpretación de la imagen el que tenga mayor probabilidad. Para algunos casos la estructura de las red puede ser un árbol o conjunto de árboles, en cuyo caso la propagación de probabilidades es muy rápida. Para el caso general, la estructura es una red multiconectada, en la cual, si el proceso de cálculo es más complejo.

Esta representación ha sido aplicada en reconocimiento en varios dominios, entre ellos para partes industriales, identificación de barcos y análisis de imágenes para endoscopía.

El endoscopio es un instrumento que se utiliza para observar el interior del tubo digestivo (ver figura 11.8). El caso de endoscopía, se utiliza una RP para representar los diferentes objetos de interés en imágenes del interior del tubo digestivo. La estructura de la red bayesiana obtenida para esto dominio se muestra en la figura 11.9. En base a esta estructura se pueden reconocer los diferentes tipos de objetos (nodos superiores), en base a las características obtenidas de la imagen (nodos inferiores), mediante la propagación de probabilidades de abajo hacia arriba. Dicho proceso de propagación, obtiene la probabilidad posterior de cada objeto (lumen, diverticulo, etc.), pudiendo entonces seleccionar el objeto de mayor probabilidad.

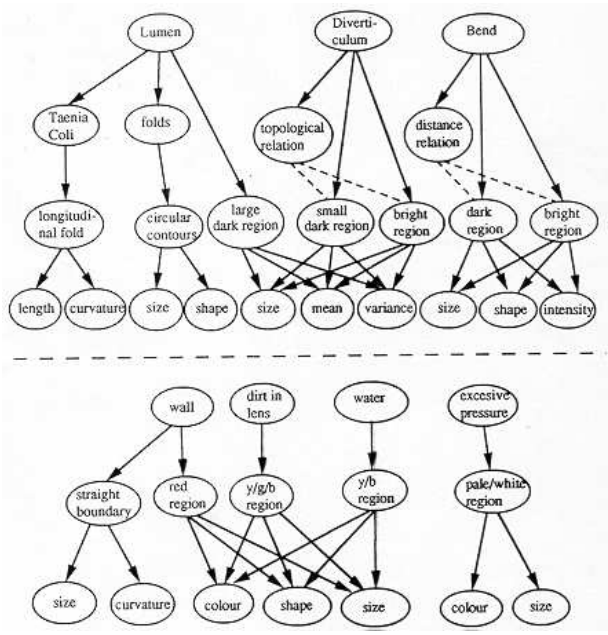


Figura 11.9: Estructura de una RP para el reconocimiento de objetos en imágenes de endoscopía.

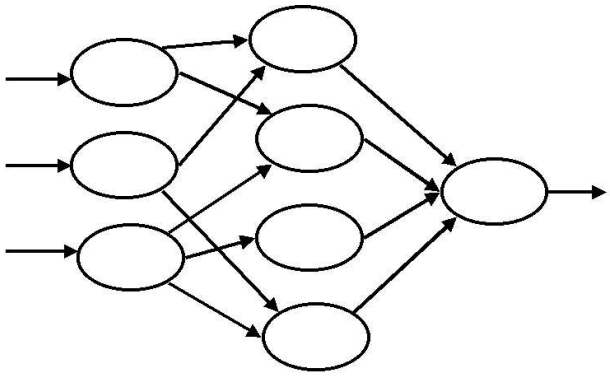


Figura 11.10: Red neuronal.

11.8 Redes neuronales

Una red neuronal es un conjunto de procesadores muy simples (neuronas) interconectados que forman lo que se considera un modelo simplificado del cerebro. Una neurona artificial tiene, generalmente, varias entradas y una salida. La salida es una función de la suma de la entradas, multiplicadas por "pesos" asociados a las interconexiones entre neuronas:

$$O = f(\sum_i W_i I_i) \tag{11.2}$$

Donde O es la salida, I_i son las entradas y W_i los pesos asociados.

Dichas neuronas se encuentran interconectadas formando una "red neuronal" (ver figura 11.10). Algunas tienen iterconexiones al mundo externo (entrada / salida) y otras son internas (escondidas). Las redes neuronales se utilizan, normalmente, como elementos clasificadores o memorias asociativas, asociando una serie de patrones de entrada con patrones de salida. Para ello se "entrena" la red, alterando los pesos de las interconexiones de acuerdo a la relación deseada.

Figura 11.11: Imágenes a diferentes resoluciones (estructura piramidal) utilizadas para reconocimiento de ojos en caras humanas con redes neuronales.

En el caso más simple, con una red de una sóla capa (Perceptron), los pesos se ajustan para minimizar el error de salida:

$$e = O_{deseada} - O_{actual} = 0 - \sum_i W_i I_i \tag{11.3}$$

Considerando que se tiene un umbral de cero. Entonces los pesos de alteran para minimizar el error:

$$W_i(t+1) = W_i(t) + \rho I_i, e > 0 \tag{11.4}$$

Existen varios tipos de redes neuronales de acuerdo a su topología y el algoritmo de aprendizaje utilizado. Entre las más comunes están: Perceptrón, WISARD, BAM, redes Kohonen, máquinas de Boltzman, ART, y retropropagación.

11.8.1 Reconocimiento de objetos mediante redes neuronales

Una forma de aplicar las redes neuronales para reconocimiento es mediante su aplicación a ventanas de pixels en forma análoga a las máscaras para detección de orillas. Primero se entrenan con varios ejemplos positivos y negativos de los objetos de interés, y después se aplican a toda la imagen detectando la localización de dichos objetos donde se tenga mayor respuesta. Esto se puede extender a diferentes resoluciones para hacer el proceso más eficiente. Para ello se utiliza una estructura piramidal para representar la imagen, ver figura 11.11, y se comienza por los niveles superiores (menor resolución), pasando al siguiente nivel cuando exista cierta respuesta, hasta llegar a la máxima resolución. El proceso de entrenamiento se puede optimizar mediante el mapeo de los pesos de las redes de ciertas resoluciones a otras.

Ésta idea ha sido aplicada al reconocimiento de ojos en caras humanas usando una red tipo retropropagación. El problema con este enfoque es que la red NO es invariante ante cambios de escala y rotación. Otra alternativa, más promisoria, es utilizar características obtenidas de los niveles bajo e intermedio como entradas a la red neuronal.

11.9 Referencias

Existen múltiples trabajo sobre el enfoque basado en conocimiento para visión, y sigue siendo un área activa de investigación. Rao [96] presenta un análisis general sobre representación y control en visión. El sistema VISIONS fué desarrollado por Hanson [32]. Ballard desarrolló uno de los primeros sitemas sistemas basados en redes semánticas [2].

Sobre la visión basada en conocimiento también puede citarse el trabajo de Nazif y Levine [82] como un ejemplo de sistema basado en reglas (ver capítulo de segmentación). Otro sistema experto interesante es el de Fischler y Strat [22], quienes reconocían árboles en escenas naturales. Las hipótesis las generaba a partir del follaje y tronco. Stenz [110] describe el sistema CODGER que se utilizó para el proyecto NavLab, CODGER esta basado en reglas y una arquitectura de pizarrón para compartir la información de los diferentes módulos. Todos los anteriores sistemas estan basados en reglas, las cuales se construyeron de manera empírica. Una metodologí, basada en aprendizaje de máquina, ha sido propuesta por R. Michalski [55, 56] para "descubrir" los atributos involucrados en las reglas.

Una introducción general sobre redes bayesianas se puede consultar en el libro de Pearl [88]. El uso de redes bayesianas en visión fue inicialmente propuesto por Levitt y Binford [67] y por Sucar y Gillies [112]. La aplicación a endoscopía se describe en [112, 111].

Las redes neuronales se han utilizado extensamente en reconocimiento de patrones, el lector interesado puede consultar los *surveys* [108] y [33]. Vease también los trabajos de T. Kanade [102] y [100]. El enfoque multi–resolución para reconocimiento de ojos se describe en [31].

Los crierios para representaciones para visión fueron propuestos por [53].

11.10 Problemas

1. Cuál es la diferencia fundamental entre visión basada en modelos geom'etricos vs. visión basada en conocimiento? Para qué tipo de dominios y aplicaciones es más adecuado cada enfoque?

2. Plantea una representación en base a (a) reglas, (b) redes semánticas/marcos y (c) redes probabilísticas para reconocer visualmente mesas y sillas.

3. Que tipo de preprocesamiento se puede aplicar a una imagen (sin realizar segmentación) antes de aplicar una red neuronal, para evitar los problemas de escalamiento y rotación.

4. Dada una red probabilística de sólo dos niveles (1 objeto y n atributos), dar una expresión para obtener la probabilidad posterior del objeto dados los atributos en base al teorema de Bayes.

5. Se desea implementar un sistema de visión que reconozca diversas clases de frutas. Describe la parte de alto nivel en base a (a) reglas de producción, (b) prototipos (frames), y (c) redes probabilísticas.

Bibliografía

[1] T. Alter, R. Basri, *Extracting Salient Curves from Images: An Analysis of the Saliency Network*, IJCV, vol 27(1), pp. 51-69, Marzo 1998.

[2] D. Ballard, C. Brown, *Computer vision*, New Jersey: Prentice Hall, 1982.

[3] D. Ballard, *Generalizing the Hough Transform to Detect Arbitrary Shapes*, Pattern Recognition, vol. 13(2), pp. 111-122, 1981.

[4] R. Bajcsy, *Computer Description of Textured Surfaces. Proceedings International Conference on Artificial Intelligence*, Stanford, Calif., pp. 572–579, 1973.

[5] F. Bergholm, *Edge focusing.* IEEE Trans. on PAMI, vol. 9(6), pp. 726-741, noviembre 1987.

[6] P. Besl, R. Jain, *Three-dimensional object recognition*, ACM Compu. Surveys, vol. 17(1), pp. 75-145, 1985.

[7] C. R. Brice, C. L. Fennema, *Scene analysis using regions*, Artificial Intelligence, vol. 1(3), pp. 205-226, 1970.

[8] P. Brodatz, *Textures: A photographic album for art and designers.* New York: Dover Publications, Inc., 1966.

[9] R. Brooks, *Model-Based 3-D Interpretation of 2-D Images*, IEEE Trans. on PAMI, vol. 5(2), pp. 140-149, March, 1983.

[10] J. Canny, *A computational approach to edge detection.* IEEE Trans. on PAMI, vol. 8(6), pp. 679-698, noviembre 1986.

[11] K. Castleman, *Digital image processing*, New Jersey: Prentice Hall, 1996.

[12] R. Chellappa, S. Chatterjee, *Classification of Textures Using Gaussian Markov Random Fields*, IEEE Trans. on ASSP, vol. 33, pp. 959-963, August 1985.

[13] B. S. Manjunath, R. Chellappa, *Unsupervised Texture Segmentation Using Markov Random Field Models*, IEEE Trans. on PAMI, vol. 13(5), pp. 478-482, Mayo 1991.

[14] J. Chen, P. Saint-Marc, G. Medioni, *Adaptive smoothing: a general tool for early vision.* Proc. of the Int. Conf. on CVPR, pp. 618-624, 1989.

[15] C. K. Chow, T. Kaneko, *Boundary detection of radiographic images by a threshold method*, Proc. of IFIP Congress, pp. 130-134, 1971.

[16] M. Clowes, *On seeing things*, Artificial Intelligence, vol. 2(1), pp. 79-116, 1971.

[17] E. R. Davies, *Machine vision.* London: Academic Press, 1997.

[18] R. Duda, P. Hart, *Use of the Hough Transform to Detect Lines and Curves in Pictures*, Comm. of ACM, vol 15(1), pp. 11-15, Jan. 1972.

[19] R. Duda, P. Hart, *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons, 1973.

[20] J. Elder, S. Zucker, *Local scale control for edge detection and blur estimation*. IEEE Trans. on PAMI, vol. 20(7), pp. 699-716, julio 1998.

[21] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, Cambridge, MA: MIT Press, 1993.

[22] M. A. Fischler, T. M. Strat, *Recognizing objects in a natural environment: a contextual vision system (CVS)*, Proc. of the Image Understanding Workshop, DARPA, pp. 774-796, May 1989.

[23] J.D. Foley, A. Van Dam, *Fundamentals of interactive computer graphics*. Reading, Mass.: Addison–Wesley, 1982.

[24] W. Frei, C. C. Chen, *Fast boundary detection: a generalization and a new algorithm*, IEEE Trans. on Computers, vol. 26(2), pp. 988-998, Oct. 1977.

[25] J. J. Gibson, *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin, 1979.

[26] G. Gómez, *Local smoothness in terms of variance: the adaptive Gaussian filter*, Proc. of BMVC, vol. 2, pp. 815-824, 2000.

[27] G. Gómez, J.L. Marroquín, L.E. Sucar, *Probabilistic estimation of local scale*, Proc. of ICPR, vol. 3, pp. 798-801, 2000.

[28] R. González, R. Woods, *Tratamiento digital de im'agenes*. Wilmington, Delaware: Addison-Wesley Iberoamericana, 1996.

[29] A. Guzán, *Decomposition of a Visual Scene into Three-Dimensional Bodies*, AFIPS Fall Join Conferences, pp. 291-304, December 1968.

[30] J. Hadamard, *Lectures on the Cauchy problems in lineal partial differential equations*. New Haven: Yale University Press, 1923.

[31] Hand et al. *A neural network feature detector using a multi-resolution pyramid*, en Neural Networks for Vision, Speech and Natural Language, R. Linggard, D.J. Myers, C. Nightingale (eds.), Chapman & Hall, 1992.

[32] A. R. Hanson, E. M. Riseman, *The VISIONS Image-Understanding System*, Advances in Computer Vision, vol. I, pp. 1-114, 1988.

[33] J. Heikkonen, A. Bulsari (eds.), *Special Issue on Neural Networks for Computer Vision Applications*, Pattern Recognition Letters, vol 17(4), pp. 317-429, Apr. 1996.

[34] F. Heitger, R. von der Heydt, E. Peterhans, L. Rosenthaler, O. Kübler, *Simulation of neural contour mechanisms: representing anomalous contours*, IVC, vol 16 (6-7), pp. 407-421, May 1998.

[35] G. Haley, B. Manjunath, *Rotation invariant texture classification using the modified Gabor filters*, Proc. of ICIP, vol 1, pp. 262-265, octubre 1995.

[36] D. A. Huffman, *Impossible Objects as Non-Sense Sentences*, in R. Meltzer and D. Michie (eds.) Machine intelligence 6, Elsevier, pp. 295-323, 1971.

[37] S. Horowitz, T. Pavlidis, *Picture Segmentation by a Directed Split and Merge Procedure*, Proc. of the ICPR, pp. 424-433, 1974.

[38] B. K. P. Horn, *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*, MIT AI TR-232, jun. 1970 (PhD thesis).

[39] B. K. P. Horn, *Obtaining Shape from Shading Information*, en P. H. Winston (ed.), The Psychology of Computer Vision, pp. 115-155, New York: McGraw-Hill, 1975.

[40] B. Horn, B. Schunk, *Determining optical flow: a retrospective*. Artificial Intelligence, artint 1000, vol. 59, pp. 81-87, 1993.

[41] P. Hough, *Method and means for recognising complex patterns*. US Patent 3069654, 1962.

[42] D. P. Huttenlocher, S. Ullman, *Object Recognition Using Alignment*, Proc. of the ICCV, pp. 102-111, 1987.

[43] L. Itti, C. Koch, E. Niebur, *A model of saliency-based visual attention for rapid scene analysis*. IEEE Trans. on PAMI, vol. 20(11), pp. 1254-1259, noviembre 1998.

[44] M. Jägersand, *Saliency maps and attention selection in scale and spatial coordinates: an information theoretic approach*. Proc. of the ICCV, pp. 195-202, 1995.

[45] M.J. Jones, J.M. Regh, *Statistical color models with application to skin detection*. Proc. of the CVPR, vol. I, pp. 274-280, 1999.

[46] B. Julesz, *Texture and visual perception*, Sci. American, vol. 212, pp. 38-48, 1965.

[47] B. Julesz, *Experiments in the Visual Perception of texture*. Sci. American, vol. 232(4), pp. 34-43, April 1975.

[48] B. Julesz, *Textons, the elements of texture perception and their interactions*, Nature 290, pp. 91-97, 1981.

[49] H. Kalviainen, P. Hirvonen, L. Xu, E. Oja, *Probabilistic and Nonprobabilistic Hough Transforms: Overview and Comparisons*, IVC vol. 13(4), pp. 239-252, May 1995.

[50] T. Kanade, *A Theory of the Origami World*, Artificial Intelligence, vol. 13(3), pp. 279-311, 1980.

[51] T. Kanade, *Recovery of the Three-Dimensional Shape of and Object from a Single View*, Artificial Intelligence, vol 17, pp. 409-460, 1981.

[52] T. Kao, S. Horng, Y. Wang, K. Chung, *A Constant Time Algorithm for Computing Hough Transform*, Pattern Recognition, vol. 26, pp. 277-286, 1993.

[53] A. K Mackworth , *Adequacy Criteria for Visual Knowledge Representation*, en Computational Processes in Human Vision: An Interdisciplinary Perspective, Zenon W. Pylyshyn (ed.), pp. 462-474, Ablex, 1988.

[54] J. Maeda, C. Ishikawa, S. Novianto, N. Tedehara, Y. Suzuki, *Rough and Accurate Segmentation of Natural Color Images Using Fuzzy Region-growing Algorithm*, Proc. of the ICPR, vol 3, pp. 642-645, 2000.

[55] R. S. Michalski, Q. Zhang, M. A. Maloof, E. Bloedorn, *The MIST Methodology and its Application to Natural Scene Interpretation*, Proceedings of the Image Understanding Workshop, Palm Springs, CA, pp. 1473-1479, Feburary, 1996.

[56] R. S. Michalski, A. Rosenfeld, Z. Duric, M. A. Maloof, Q. Zhang, *Learning Patterns in Images*, in Michalski, R.S., Bratko, I. and Kubat, M. (Eds.), Machine Learning and Data Mining: Methods and Applications, London: John Wiley & Sons, pp. 241-268, 1998.

[57] W. Niblack, *An introduction to Digital Image Processing*, pp. 115-116, Englewood Cliffs: Prentice Hall, 1986.

[58] G. Khan, D. Gillies, *Extracting Contours by Perceptual Grouping*. Image and Vision Computing, vol. 10(2), pp. 77-88, 1992.

[59] G. Khan, D. Gillies, *Parallel-Hierarchical Image Partitioning and Region Extraction*, In L. Shapiro, A. Rosenfeld (ed.) Computer Vision and Image Processing, Boston: Academic Press, pp. 123-140, 1992.

[60] N. Kiryati, Y. Eldar, A. Bruckstein, *A Probabilistic Hough Transform*, Pattern Recognition, vol. 24(4), pp. 303-316, 1991.

[61] R. A. Kirsch, *Computer determination of the constituents structure of biological images*, Computers and Biomedical Research, vol. 4(3), pp. 315-328, Jun. 1971.

[62] M. S. Kiver *Color television fundamentals*, New York: McGraw–Hill, 1965.

[63] C. Koch and S. Ullman, *Shifts in selective visual attention: towards the underlying neural circuitry*, Human Neurobiology, vol. 4, pp. 219-227, 1985.

[64] J. J. Koenderink, *The structure of images*. Biological Cybernetics, vol. 50, pp. 363-370, 1984.

[65] I. Kovács, P. Kozma, A. Fehér, G. Benedek, *Late maturation of visual spatial integration in humans*, Proc. Natl. Acad. Sci. USA, vol. 96(21), pp. 12204-12209, Oct. 1999.

[66] J.S. Levine, E.F. MacNichol, *Color Vision in Fishes*. En *The Mind's Eye*, Readings from Scientific American, New York: W.H. Freeman, 1986.

[67] T. O. Binford, T. S. Levitt, W. B. Mann, *Bayesian Inference in Model-Based Machine Vision*, Uncertainty in AI, vol. 3, 1989

[68] Z. Li, B. Yao, F. Tong, *Linear Generalized Hough Transform and Its Parallelization*, IVC vol. 11, pp. 11-24, 1993.

[69] T. Lindeberg, *Edge detection and ridge detection with automatic scale selection*. IJCV, vol. 30(2), pp. 117-154, 1998.

[70] J. Liter, H. H. Bülthoff, *An Introduction to Object Recognition*, Technical report 43, Max Planck Institute - Tübingen, Nov. 1996.

[71] D. Lowe, *Perceptual Organization and Visual Recognition*, Boston: Kluwer Academic Publishers, 1985.

[72] S. Y. Lu, K. S. Fu, *A Syntactic Approach to Texture Analysis*, CGIP, vol 7(3), pp. 303-330, June 1978.

[73] S. Y. Lu, K. S. Fu, *Stochastic Tree Grammar Inference for Texture Synthesis and Discrimination*, CGIP, vol. 9, pp. 234-245, 1979.

[74] Y. Lu, C. Jain, *Reasoning about edges in scale space*. IEEE Trans on PAMI, vol. 14(4), pp. 450-468, abril 1992.

[75] A. Martínez, *Navegación Robótica basada en Forma de Sombreado*, Tesis de Maestría, ITESM Campus Morelos, 1996.

[76] D. Marr, E. Hildreth, *Theory of edge detection*. Proc. of the Royal Soc. of London, vol. B-207, pp. 187-217, 1980.

[77] D. Marr, *Vision*. San Francisco: Freeman, 1982.

[78] J. Matas, C. Galambos, J. Kittler, *Progressive Probabilistic Hough Transform for Line Detection*, Proc. of CVPR, vol. 1, pp. 554-560, 1999.

[79] S. M. Menegos, *Edge Detection in Color Images*. Tesis de Maestría. Departamento de Computación, Imperial College, Londres, 1992.

[80] A. Moghaddamzadeh, N. Bourbakis, *A Fuzzy Region Growing Approach for Segmentation of Color Images*, Pattern Recognition, vol. 30(6), pp. 867-881, june 1997.

[81] K. Nakayama, G. H. Silverman, *Serial and Parallel Processing of Visual Feature Conjunctions*. Nature, vol. 320, pp. 264-265, 1986.

[82] Nazif, Levine, *Low Level Image Segmentation: An Expert System*, IEEE Trans. on PAMI, vol. 6(5), pp. 555-577, Sep. 1984.

[83] W.S. Ng, C. Lee, *Comment on Using the Uniformity Measure for Performance-Measure in Image Segmentation*, IEEE Trans. on PAMI, vol. 18(9), pp. 933-934, Sep. 1996.

[84] Ø. Trier, A. Jain, *Goal-Directed Evaluation of Binarization Methods*, IEEE Trans. on PAMI, vol 17(12), pp. 1191-1201, Dec. 1995.

[85] A. Papoulis, *The Fourier Integral and its Applications*. New York: McGraw–Hill, 1962.

[86] J. R. Parker, *Algorithms for image processing and computer vision*. New York: John Wiley & Sons, Inc., 1997.

[87] T. Pavlidis, *Comments on "Low Level Image Segmentation: An Expert System"*, IEEE Trans. on PAMI. vol 8(5), pp. 675-676, Sep. 1986.

[88] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufmann, 1988

[89] P. Perona, J. Malik, *Scale space and edge detection using anisotropic diffusion*. IEEE Trans. on PAMI, vol. 12(7), pp. 629-639, julio 1990.

[90] P. Perona, T. Shiota, J. Malik, *Anisotropic diffusion*, in B. M. ter Haar Romeny (ed.), Geometry-driven diffusion in computer vision, pp. 72-92. Dordrecht: Kluwer Academic Publishers, 1994.

[91] J. Piaget, B. Inhelder, *L'Image mentale chez l'Enfant*, Presses Universitaire de France, 1966.

[92] J. M. Prager, *Extracting and labeling boundary segments in natural scenes*, IEEE Trans. on PAMI, vol. 2(1), pp. 16-27, 1980.

[93] J. M. S. Prewitt, *Object enhancement and extraction*, in B.S. Lipkin and A. Rosenfeld (eds.), Picture processing and psychopictorics, pp. 75-149, New York: Academic Press, 1970.

[94] V. I. Ramachandran, *Perceiving Shape From Shading*, Scientific American, vol. 259(2), pp. 76-83, 1988.

[95] T. Randen, J. H. Husy, *Filtering for texture classification: A comparative study*. IEEE Trans. on PAMI, vol. 21(4), pp. 291-310, abril 1999.

[96] A. R. Rao, R. Jain *Knowledge representation and control in computer vision systems*, IEEE Expert, Vol. 3(1), pp. 64-79, Spring 1988.

[97] H. Rashid, P. Burguer, *Differential Algorithm for the Determination of Shape from Shading Using a Point Light Source*, Image and Vision Computing, vol 10(2), pp. 119-127, 1992.

[98] L. G. Roberts, *Machine perception of three-dimensional solids*, in J. Tippett et al. (eds.), Optical and electro-optical information processing, pp. 159-197, Cambridge, MA: MIT Press, 1965.

[99] P. Saint-Marc, J. Chen, G. Medioni, *Adaptive smoothing: a general tool for early vision*. IEEE Trans. on PAMI, vol. 13(6), pp. 514-529, junio 1991.

[100] H. Rowley, S. Baluja, T. Kanade, *Neural Network-Based Face Detection*, IEEE Trans. on PAMI, vol 20(1), pp. 23-38, Jan. 1998.

[101] M. Sato, S. Lakare, M. Wan, A. Kaufman, *A Gradient Magnitude Based Region Growing Algorithm for Accurate Segmentation*, Proc. of the ICIP, 2000.

[102] S. Satoh, T. Kanade, *Name-It: Association of Face and Name in Video*, Proc. of the CVPR, pp. 368-373, 1997.

[103] C.E. Shannon, *A mathematical theory of communication*. Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, julio y octubre 1948.

[104] A. Shashua, S. Ullman, *Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network*, Proc. of the ICCV, vol 1, pp. 321-327, 1988.

[105] J. Shen, S. Castan, *An optimal linear operator for step edge detection*, CVGIP: Graphical models and understanding, vol. 54(2), pp. 112-133, 1992.

[106] J. Shi, J. Malik, *Normalized Cuts and Image Segmentation*, IEEE Trans. on PAMI, vol. 22(8), pp. 888-905, Aug. 2000.

[107] Shirai, *Recognition of Real World Objects using Edge Cue*, In Hanson y Riseman (eds.) Computer Vision Systems, New York: Academic Press, 1978. (*verificar referencia*)

[108] J. Skrzypek, W. Karplus (eds.), *Special Issue-Neural Networks in Vision and Pattern Recognition*, Int. Journal of Pattern Recognition and Artificial Intelligence, vol 6(1), pp. 1-208, Apr. 1992.

[109] S. Smith, J.M. Brady, *SUSAN - A new approach to low level image processing*. IJCV, vol. 23(1), pp 45-78, mayo 1997.

[110] A. Stenz, *The NAVLAB System for Mobile Robot Navigation*, CMU Technical Report CMU-CS-90-123, 1990.

[111] L. E. Sucar, D. F. Gillies, *Handling Uncertainty in knowledge-based computer vision*, en Symbolic and Quantitative Approaches to Uncertainty, Springer-Verlag: LNCS 548, R. Kruse and P. Siegel (eds.), pp. 328-332, 1991.

[112] L.E. Sucar, D.F. Gillies y H. Rashid, *Integrating Shape from Shading in a Gradient Histogram and its Application to Endoscope Navigation*, International Symposium on Artificial Intelligence, AAAI Press, pp. 132–139, 1992.

[113] L.E. Sucar, A. Martínez, *Navegación robótica basada en forma de sombreado*, Computación Visual 97, México, pp. 193–199, 1997.

[114] G. Kanizsa, *Subjective Contours*, Scientific American, vol. 234(4), Apr. 1976.

[115] G. Kanizsa, *Organization in Vision: Essays on Gestalt Perception*, New York: Praeger, 1979.

[116] H. Samet, *The Quadtree and Related Hierarchical Data Structures*, ACM Computing Surveys, vol. 6(2), pp. 187-260, June 1984.

[117] E. S. Spelke, *Origins of Visual Knowledge*, In An Invitation to Cognitive Science. Vol 2. Ed. D. Osherson, S. Kosslyn and J. Hollerbach. pp. 99-127. Cambridge, MA: MIT Press, 1990.

[118] F. Tomita, S. Tsuji, *Computer analysis of visual textures*. Norwell, Mass: Kluwer Academic Publishers, 1990.

[119] E. Trucco, A. Verri, *Introductory Techniques for 3-D Computer Vision*, New York: Prentice Hall, 1998.

[120] V. Torre, T. Poggio, *On edge detection*, IEEE Trans. on PAMI, 8(2): 147-163, Mar. 1986.

[121] J. Tou, R. Gonzalez, *Pattern Recognition Principles*, Reading: Addison-Wesley, 1974.

[122] A. Treisman, G. Gelade, *A feature integration theory of attention*. Cognitive Psychology, vol. 12, pp. 97-136, 1980.

[123] A. Treisman, J. Souther, *Illusory Words: The Roles of Attention and Top-Down Constraints in Conjoining Letters to Form Words*". Journal of Experimental Psychology: Human Perception and Performance, vol. 14, pp. 107-141, 1986.

[124] A. Treisman, S. Gormican, *Feature analysis in early vision: Evidence from search asymmetries*. Psychological Review, vol. 95, pp. 15-48, 1988.

[125] T. Tuytelaars, M. Proesmans, L. Van Gool, *The Cascaded Hough Transform*, Proc. of the ICIP, vol. 2, pp. 736-739, october 1997.

[126] S. Ullman, *The Interpretation of Visual Motion*, Cambridge: MIT Press, 1979.

[127] S. Ullman, *Visual Routines*, Cognition, vol 18, pp. 97-156, 1984.

[128] S. Ullman, *An Approach to Object Recogniton: Aligning Pictorial Descriptions*, Cognition, vol. 32, pp. 193-254, 1986.

[129] S. Ullman, R. Basri, *Recognition by Linear Combinations of Models*, IEEE Trans. on PAMI, vol. 13(10), pp. 992-1006, October 1991.

[130] S. Ullman, *High-level vision*. Cambridge: MIT Press, 1996.

[131] P. Suetens, P. Fua, A. J. Hanson, *Computational strategies for object recognition*, ACM Comp. Surveys, Vol. 24(1), pp. 5-62, 1992.

[132] K. Vincken, A. Koster, M. Viergenver. *Probabilistic multiscale image segmentation*. IEEE Trans. on PAMI, vol. 19(2), pp. 109-120, febrero 1997.

[133] D. L. Waltz, *Generating semantic description from drawings of scenes with shadows*, Artificial Intelligence, vol. 2, pp. 79-116, 1971.

[134] J. Weickert, "A review of nonlinear diffusion filtering". In B. ter Haar Romeny, L. Florack, J. Koenderink, M. Viergever (Eds.), *Scale-Space Theory in Computer Vision*, Berlin: Springer-Verlag, LNCS 1252, pp. 3-28, 1997.

[135] J. Weickert, "Coherence-enhancing diffusion filtering". *IJCV*, vol. 31, pp. 111-127, 1999.

[136] A.P. Witkin, *Scale-space filtering*. Proc. of the IJCAI, vol 2, pp 1019-1022, agosto 1983.

[137] J. M. Wolfe, K. R. Cave, S. L. Franzel, *Guided Searc h: An Alternative to the Feature Integration Model for Visual Search*. Journal of Experimental Psychology: Human Perception and Performance, vol. 15(3), pp. 419-433, 1989.

[138] S. Yuen, C. Ma, *An Investigation of the Nature of Parameterization for the Hough Transform*, Pattern Recognition, vol. 30(6), pp. 1009-1040, June 1997.