Índice General

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción a la visión artificial

Introducción y orientaciones para el estudio

Uno de los procesos de percepción más importantes en los seres biológicos es la visión ya que proporciona el conjunto de información más extenso sobre el entorno. Como las técnicas y elementos involucrados en el proceso de visión son los más importantes y extensos, prácticamente toda la parte de percepción de esta asignatura está dedicada a la visión por computador. La visión por computador implica el desarrollo de estrategias computacionales que intenten modelar ciertos atributos de la percepción visual humana teniendo en cuenta las restricciones físicas del hardware existente. Se trata de un objetivo extremadamente ambicioso estando actualmente en una etapa muy primitiva de desarrollo.

En esta unidad, además de definir la terminología utilizada habitualmente en un sistema de visión por computador, intentaremos mostrar al alumno la grandeza del problema recalcando la necesidad de su descomposición en diferentes subtareas cuyo propósito particular será definido de forma muy breve en esta unidad y más ampliamente en las unidades sucesivas. La idea es que el alumno piense en un sistema de visión como la ejecución de un conjunto de etapas interrelacionadas que sirven para resolver un determinado problema (determinar el número de glóbulos rojos presentes en una imagen, reconocer un objeto, etc.). Otro objetivo fundamental de este tema es que el alumno perciba la necesidad de inyectar conocimiento en cada una

de las etapas de procesado para poder llegar a una solución del problema de visión planteado sin incertidumbre. Finalmente se dedicaran dos secciones a describir los componentes básicos de un sistema de visión por computador y a mostrar al alumno algunas aplicaciones halladas en la industria en donde la visión por computador juega un papel importante.

Tanto para el alumno de Ingeniería de Sistemas como de Gestión este capítulo es completamente nuevo. Dado su carácter introductorio, la forma de estudio recomendable es la lectura detallada de su contenido.

Objetivos

Comprender la complejidad del problema de visión por computador, conocer las distintas subtareas en las que se puede descomponer y percibir la necesidad de inyectar conocimiento durante el proceso. Este conocimiento permitirá al alumno seleccionar el conjunto de subtareas más adecuado para abordar un determinado problema.

1.1 Conceptos generales.

El éxito en la operación de cualquier máquina inteligente depende de su habilidad para hacer frente a eventos inesperados. Estas máquinas cognitivas deben percibir, memorizar y comprender el entorno, constantemente cambiante, en el que trabajan. Según [11], la IA contempla el estudio de las ideas que permiten a las máquinas comportarse como seres inteligentes. Un término difícil de definir es el comportamiento inteligente. Una alternativa consistiría en describir una gran serie de habilidades ligadas al comportamiento inteligente tales como la habilidad de razonar, adquirir y aplicar el conocimiento, la facultad de percibir y manipular cosas del mundo físico, la facultad de combinar reglas de actuación etc. Para lograr este fin los seres inteligentes disponen de diferentes medios sensoriales cuya función es suministrar información sobre su entorno que permita guiar su comportamiento.

Desde este punto de vista podemos definir la percepción como el conjunto de actividades que proporcionan al ser vivo la información requerida del medio. Para ello el ser inteligente dispone de una serie de sensores que pueden ser de diferentes tipos: de tacto, quimicos, ultrasonidos, infrarojos, eléctrico o visuales entre otros. Dependiendo de las características del medio será más conveniente el uso de uno u otro tipo de sensor. Así, por ejemplo, en ausencia de luz (charcas turbias o fondo submarino) puede resultar conveniente la utilización de sensores de campo eléctrico. No obstante, entre los sistemas sensoriales que dispone el ser vivo, son los sensores visuales los que mayor información dan del entorno (color, textura, forma) pero, por contra son los más complejos.

La percepción en los seres vivos ha evolucionado con el aumento de la complejidad de las especies y por tanto, con la cantidad de información necesaria para su supervivencia. Esto es, la cantidad de información que el ser vivo requiere de su sistema perceptual es dependiente de la cantidad de acciones que dicho ser vivo es capaz de realizar. Evidentemente, si el número de comportamientos es pequeño, serán simples los requisitos impuestos al sistema perceptual. Por ejemplo, en el caso de la lombriz de tierra únicamente se exige a su sistema perceptual que detecte la presencia o ausencia de luz para lo cual le basta disponer de unos fotoreceptores en forma de lente en la epidermis. Sin embargo, si subimos en la jerarquia animal y analizamos un animal en donde se aprecia mayor complejidad de comportamientos observamos también sistemas visuales más complejos. Un ejemplo es el sistema de captura/huida de la rana pipiens. En esta ocasión el comportamiento de la rana podría resumirse como: cuando detecta un objeto muy pequeño lo clasifica como insecto (alimento) y cuando detecta un objeto grande lo identifica como enemigo y emprende la huida. En definitiva hemos observado que no sólo aumenta la complejidad de los organos sensoriales al aumentar el número de comportamientos sino que vemos que aparecen representaciones internas de lo percibido.

Pues bien, si seguimos subiendo en la jerarquía animal y analizamos al ser humano observamos que aumenta tanto la cantidad como la complejidad de las acciones

posibles, con lo cual se necesita más información para su selección y en consecuencia aumenta la complejidad de los órganos sensoriales independizandose la percepción de la acción. La respuesta se basa cada vez más en la representación interna que el animal genera a partir de lo percibido y de su conocimiento previo. Dicha representación interna puede depender directamente de la entrada, en cuyo caso tenemos un modelo lineal de procesado de información, en el que la percepción controla totalmente la acción, o bien que el conocimiento que ya posee el individuo influya y controle la propia percepción. En definitiva, se genera un modelo de la situación a partir de lo percibido y del conocimiento existente. Es decir, en los seres biológicos de mayor nivel en la jerarquía animal es clara la aparición de una tarea intermedia entre percepción y acción.

De todo lo dicho podemos concluir diciendo que el objetivo principal de la los sistemas sensoriales en los animales superiores, en el hombre y en las máquinas inteligentes es proporcionar información de la escena orientada a la tarea que se está desarrollando que permita decidir la acción más conveniente tanto en tiempo como en forma.

Como hemos dicho anteriormente es la visión el más rico de los procesos sensoriales debido a su naturaleza informativa y al alejamiento de los sensores visuales de la escena física. Nuestro propio sistema visual es relativamente rápido y altamente robusto. Hasta el momento, esta sofisticada forma de visión perceptual únicamente la disfrutan los organismos biológicos de alto orden, tales como los seres humanos. Estos atributos básicos todavía no se encuentran en los algoritmos de visión por computador y en los sistemas hardware que han sido desarrollados a lo largo de los años para las aplicaciones en máquinas de visión. Por ese motivo, en esta asignatura dedicamos la parte de percepción a los sensores visuales dado que es clara candidata a la aplicación de técnicas de IA. No obstante, combiene aclarar que muchas de las técnicas aplicadas a la visión por computador son igualmente aplicables a los datos procedentes de otro tipo de sensores.

La visión artificial implica el desarrollo de estrategias computacionales que intenten modelar ciertos atributos de la percepción visual humana teniendo en cuenta las restricciones físicas del hardware existente. Tales modelos son prerequisitos necesarios para el reconocimiento e interpretación de una escena que permita emitir juicios en torno a objetos. Un análisis detallado de la percepción visual nos revela numerosas complejidades y dificultades en la percepción del mundo físico tridimensional a través de sus proyecciones bidimensionales. Nosotros percibimos el mundo de objetos 3D con muchas propiedades invariantes. Los datos visuales de entrada no presentan la correspondiente invarianza, sino que contienen mucha información irrelevante. De alguna manera nuestro sistema visual, desde la retina hasta los niveles cognitivos, impone orden sobre las entradas visuales. Esto lo lleva a cabo empleando información intrínseca extraída de los datos de entrada, y también a través de consideraciones y conocimiento a piori aplicado sobre los diferentes niveles de procesamiento visual. Por tanto, se trata de un objetivo extremadamente ambicioso y complejo estando actualmente en una etapa muy primitiva de desarrollo.

La visión artificial se relaciona con diversos campos entre los que se encuentran aquellos que comparten técnicas comunes de procesamiento de datos visuales:

- **Procesamiento de imágenes digitales**, que involucra la transformación de una imagen para obtener otra de mayor calidad o con alguna característica resaltada que facilite la posterior extracción de información.
- **Reconocimiento de patrones**, que aborda la clasificación de objetos en clases representadas por patrones.
- **Inteligencia artificial** y más concretamente los problemas de interpretación, aprendizaje y razonamiento cognitivo.
- **Gráficos por computador**, cuyo objetivo es transformar a una imagen bidimensional el mundo real. Es, precisamente, el proceso inverso al que se realiza en visión artificial.

Así, cualquiera de las técnicas de procesamiento de imágenes puede ser utilizada en el diseño de los módulos de preproceso en los sistemas de visión artificial; la clasificación/reconocimiento de patrones puede entenderse como un caso especial de la visión en el sentido de que su resultado último es el nombre o categoría de un patrón de formas; o bien el problema de interpretación de imagen puede ser tratado como un problema de inteligencia artificial aplicando las técnicas estudiadas en ese campo.

Por supuesto, dependiendo del problema concreto una componente puede ser más importante que otra. Por ejemplo, en un problema de visión estéreo normalmente se involucra menos IA que en un problema de interpretación de la imagen; o bien en aplicaciones de visión como medio sensorial de un sistema robótico es de vital importancia conocer la relación geométrica entre el mundo 3D y la imagen 2D mientras que en un problema de inspección automática no es relevante dicha información mientras que ahora es fundamental el procesamiento de la imagen y el reconocimiento de regiones y formas que en ella aparecen.

En este primer capítulo de la parte de visión artificial de la asignatura se presenta una visión general resaltando las principales dificultades a abordar en la implementación de un sistema de visión.

1.2 Introducción histórica

Los primeros trabajos relacionados con la visión por computador aparecen en la década de los cincuenta. En sus comienzos se pensaba que el desarrollo del sentido artificial de la vista sería una tarea fácil y alcanzable en pocos años. Esta idea se reafirmó con el éxito conseguido por Roberts en 1965 que desarrolló programas para deducir la estructura y distribución tridimensional de varios poliedros de vértices triédricos a partir de sus imágenes digitales. En ella se detectaban los puntos de contorno para ser agrupados en segmentos más largos que idealmente correspondían

a los bordes rectos de la escena poliédrica. Cadenas cerradas de segmentos rectos formaban las caras de los bloques de la imagen. Un análisis topológico de las líneas, vértices y regiones resultantes permitía encontrar la correspondencia entre la imagen bidimensional y una estructura de datos que representaban los bloques poliédricos a partir de los cuales se construía la escena. También Wichman en 1967 levantó grandes esperanzas cuando presentó en Stanford un equipo con cámara de televisión conectada a un computador, que podía identificar objetos y sus posiciones en tiempo real. Sin embargo, y en ambos casos, se trataba de imágenes muy simples con fuertes restricciones.

Este entusiasmo inicial fue debido por un lado a una gran confianza en el poder de los computadores y por otro a la consideración de lo trivial que resulta a los seres humanos interpretar cualquier escena percibida por nuestros ojos y en consecuencia debería ser igual de fácil para los ordenadores. Sin embargo, pocos años después comenzaban a surgir frustraciones ante lo limitado de los avances obtenidos y las pocas aplicaciones existentes por lo que en los años sisuientes se abandonaron bastante las investigaciones en visión por computador. No obstante, se desarrollaron algoritmos que todavía son aplicables con éxito en algunos casos y para determinadas imágenes tales como el operador de Roberts (1965), Sobel (1970) o el de Prewitt (1970).

De este modo, a principios de los años 70, la opinión general era que la visión a bajo nivel era incapaz de producir descripciones útiles. Se observó que, de forma análoga a la aparente necesidad de la semántica en el análisis sintáctico de oraciones en inglés, un flujo de conocimientos sobre la escena hacia los niveles inferiores podía proporcionar restricciones adicionales para la interpretación. Este enfoque reducía la necesidad de cálculo a bajo nivel y enfatizaba la manipulación simbólica para la cual los ordenadores estaban más adaptados. En tales sistemas de visión dirigidos por conocimientos, se procesaban hechos sobre las posibles relaciones entre los objetos de la escena. Autores como Freude, Minsky, Shirai, Tenembaum, Barrow o Winston propusieron diferentes estructuras para efectuar esta iteración entre el procesamiento top-down y el bottom-up de la información. Para experimentarlos

se construyeron sistemas complejos que movilizaban conocimiento a todos los niveles del sistema así como información específica sobre algún dominio de aplicación. Con tal de poder completar la construcción de todos estos sistemas era inevitable realizar asunciones excesivamente simplificadoras, de manera que su eficacia seguía siendo limitada. Además, las técnicas existentes de representación y aplicación del conocimiento demostraron ser inadecuadas para superar la distancia entre la imagen de entrada y su deseada descripción simbólica.

En la década de los ochenta se empieza a hacer hincapié en la extracción de características. Así se tiene la detección de texturas (Haralik (1979)), y la obtención de la forma a través de ellas (Witkin (1981)); y ese mismo año se publican artículos sobre: visión estéreo (Mayhew y Frisby), detección del movimiento (Horn), interpretación de formas (Steven) y líneas (Kanade); o los detectores de esquinas de Kitchen y Rosendfeld (1982). Quizá el trabajo más importante de esa década es el presentado por David y sus colaboradores en 1982, donde se aborda por primera vez una metodología completa del análisis de imágenes a través del ordenador. Marr buscaba una teoría computacional de la percepción visual. El punto de partida es reconocer que cualquier explicación de la percepción visual que se base sólo en el funcionamiento de las redes neuronales desde retina a corteza será absolutamente insuficiente. Lo que necesitamos tener es una "clara comprensión de lo que se debe calcular, cómo es preciso hacerlo, los supuestos físicos en los que se basa el método y algún tipo de análisis sobre los algoritmos que son necesarios para llevar a cabo ese cálculo" [8].

De acuerdo con Marr para entender un proceso perceptual debemos comprenderlo a tres niveles:

1. NIVEL 1: Teoría computacional de la tarea a nivel genérico incluyendo cuál es el objetivo de la computación, por qué es apropiado y cuál es la lógica de la estrategia adecuada para implementarlo.

2. NIVEL 2: Representación y algoritmos: ¿cómo puede implemetarse esta

teoría de cálculo? (espacios simbólicos de representación de las entradas y salidas y algoritmos que los enlazan).

3. NIVEL 3: Implementación del algoritmo sobre una máquina paralela o serie, para tener un mecanismo operativo.

David Marr explicaba de forma clara la necesidad de inyectar conocimiento para comprender una computación en los tres niveles: "tratar de entender la percepción estudiando sólo las neuronas es como tratar de comprender el vuelo de los pájaros estudiando sólo sus plumas: simplemente, no es posible. Para comprender el vuelo de los pájaros tenemos que comprender la aerodinámica, sólo entonces nos aparece con pleno sentido la estructura de las plumas y las diferentes formas de las alas de los pájaros" [8].

Los trabajos dirigidos por Marr establecieron los fundamentos de la visión por computador moderna, trasladando el centro de atención desde las restricciones que impone el dominio de una aplicación en los sistemas de visión construidos hasta entonces, hacia las restricciones sobre las capacidades del sistema visual humano. Es decir, antes de utilizar los conocimientos sobre un dominio concreto de escenas para construir su interpretación, intentar obtener la máxima cantidad de información de las imágenes, tal como se supone que hace el sistema visual (por ejemplo somos capaces de inferir la forma de los objetos que no hemos visto nunca, por tanto sin la intervención de conocimiento de alto nivel). Información se refiere por ejemplo a la orientación y forma 3D de las superficies de los objetos que contiene la imagen, así como a parámetros físicos tales como la dirección de la luz incidente, la profundidad o la velocidad a que se desplazan los objetos de la escena. La idea es invertir el proceso físico de formación de las imágenes para reconstruir la escena a partir de ellas. A este proceso Marr lo denomina recovery, en el sentido de recuperación de la geometría de la escena.

El paso de la imagen a su descripción simbólica no es directo sino que atraviesa una representación intermedia, o quizá mejor, una jerarquia de representaciones

intermedias que van desde una representación simplificada de la imagen como una colección de cambios de intensidad hasta una representación 3D consistente en la descripción de los objetos así como sus relaciones.

De acuerdo con esta formulación la investigación se ha orientado desde entonces a descubrir y caracterizar según la teoría computacional propuesta por Marr, módulos del sistema visual humano capaces de extraer información de forma 3D a partir de las imágenes 2D. Aunque no está claro cuales son estos módulos visuales, la investigación en neurobiología, psicología y psicofísica ha encontrado importantes evidencias de que signos como el sombreado, la textura, los contornos, el movimiento y la estereovisión son muy útiles en la comprensión de las propiedades de las superficies 3D a partir de sus imágenes.

Hacia finales de los años 80 se produjo otro cambio de orientación. Los trabajos sobre módulos visuales fueron tan productivos en cuanto a resultados que la mayor parte del interés de los científicos se centró en el problema de construir sistemas de visión integrados. Esta actividad ha dado lugar a vehículos terrestres autónomos, robots, y sistemas de comprensión de imágenes de laboratorio. Ejemplos bien conocidos son los sistemas VISIONS, SIGMA o MOSAIC, cuyo objetivo no es la comprensión de un dominio concreto de escenas, sino el establecimiento de una metodología de representación del conocimiento y de razonamiento que sea extrapolable a otros dominios.

Actualmente la visión se está convirtiendo en una ciencia bien definida. La mayoría de los estudios presentados en las publicaciones de los últimos años empiezan con una descripción precisa de las representaciones sobre las que se opera o que producen los procesos visuales considerados. Las observaciones y asunciones del mundo comienzan a expresarse con sofisticados formalismos matemáticos. La detección de contornos se consigue a través de técnicas de Monte Carlo o de rigurosas teorías de regulación de la discontinuidad. Después de explicar el proceso de formación de la imagen, los investigadores se preocuparon de la geometría. La combinación de

la geometría con las propiedades de las superficies físicas conduce al análisis real multivariado y a la geometría diferencial y algebraica.

Por último reseñar una nueva línea de investigación que se ha venido a denominar la extensión del paradigma de Marr o la fusión visual: una vez establecida una teoría de la visión por computador, identificados y estudiados los diferentes módulos, el siguiente paso ha sido combinarlos entre sí para tratar de obtener o mejorar sus resultados.

1.3 Terminología

A continuación se definen una serie de conceptos básicos en visión por computador.

- **Imagen monocromática**

  Una imagen monocromática o gris se representa mediante una matriz cuyos elementos definen el valor de la intensidad del nivel de gris de cada muestra o píxel de la escena digital correspondiente. Así, el elemento $f(i, j)$ es el valor de la intensidad del nivel de gris del pixel (i, j). En la figura 1.2 se muestra un ejemplo de representación de una imagen original.

  Para poder enviar esta información al computador es necesario codificar en binario los niveles de intensidad de cada pixel. El número de bits empleados por pixel se conoce como la resolución en niveles de gris. Así por ejemplo, si se dispone de 8 bits, se tendrán 256 (2^8) niveles de gris diferentes (ver figura 1.1).

- **Imagen de color**

  - Formato RGB

    Este formato se basa en la combinación de tres señales de luminancia cromática distinta: el rojo (Red), el verde (Green) y el azul (Blue). La

manera más simple de conseguir un color concreto es determinar la cantidad de color rojo, verede y azul que se necesita combinar. Así, en una imagen RGB se define para cada canal de color una matriz de elementos de manera análoga al método empleado en la imagen gris. Un ejemplo de una imagen a color descompuesta en sus tres canales se muestra en la figura 1.2.

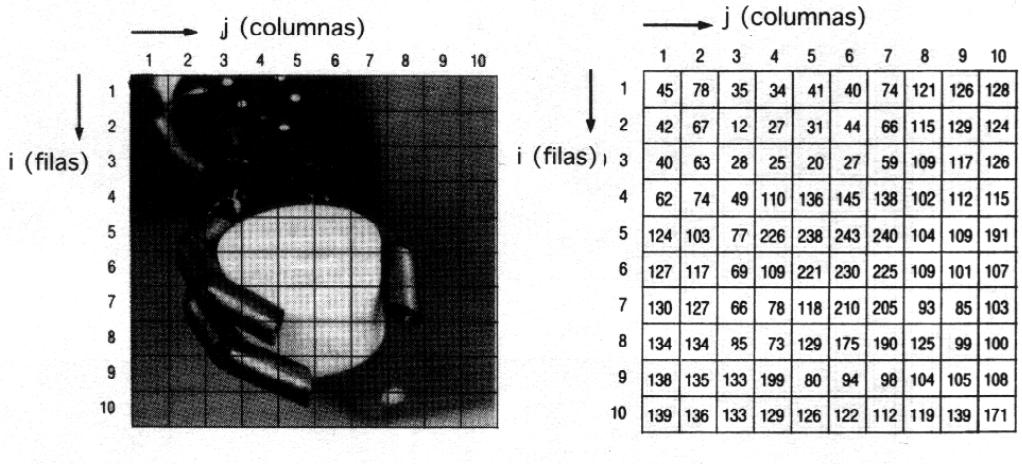
– Formato HSI

Este formato se basa en el modo de percibir los colores los seres humanos. En este caso se caracteriza el color en base a un tono, saturación y brillo. El tono representa la longitud de onda sin embargo no es igual a la longitud de onda ya que algunos colores que se encuentran en la naturaleza (el púrpura) no se encuentran en la descomposición de la luz blanca. La saturación distingue la blancura de un color y el brillo nos permite cuantificar la intensidad de dos fuentes de luz con el mismo espectro.

Las transformaciones matemáticas que permiten el paso del formato RGB al HSI se realizan mediante hardware específico:

$$\begin{aligned} \theta &= \cos^{-1}\left(\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)^{\frac{1}{2}}\right]}\right) \\ H &= \left\{\begin{array}{ll} \theta & \text{si } G \geq B \\ 2\pi-\theta & \text{en otro caso} \end{array}\right\} \\ S &= 1 - min\ (R,G,B) \\ I &= \frac{R+G+B}{3} \end{aligned}$$

donde H, S e I son el tono, saturación e intensidad de la imagen HSI resultante; y R, G y B son las componentes roja, verde y azul de la imagen RGB.



	1	2	3	4	5	6	7	8	9	10
1	45	78	35	34	41	40	74	121	126	128
2	42	67	12	27	31	44	66	115	129	124
3	40	63	28	25	20	27	59	109	117	126
4	62	74	49	110	136	145	138	102	112	115
5	124	103	77	226	238	243	240	104	109	191
6	127	117	69	109	221	230	225	109	101	107
7	130	127	66	78	118	210	205	93	85	103
8	134	134	95	73	129	175	190	125	99	100
9	138	135	133	199	80	94	98	104	105	108
10	139	136	133	129	126	122	112	119	139	171

Figura˜1.1: Imagen original y su correspondiente digitalización (Tomado de R. Ezkenazi, video signal input, Robotics Age, Age, Vol 3,N0 2, March/April 1981).

(a)Imagen color

(b) Imagen canal Rojo

(c) Imagen canal Verde

(d) Imagen canal Azul

Figura 1.2: Imagen color

- **Vecindad**

  Dado un pixel p de coordenadas (x, y) definimos la vecindad de p como el conjunto de píxeles que le rodean. Se puede distinguir:

  - *4-vecinos:* la vecindad de p está constituida por 4 vecinos situados en $\{(x+1, y),\ (x-1, y),\ (x, y+1),\ (x, y-1)\}$.
  - *8-vecinos:* la vecindad de p está constituida por 8 vecinos situados en $\left\{ \begin{array}{c} (x+1, y),\ (x-1, y),\ (x, y+1),\ (x, y-1), \\ (x+1, y+1),\ (x+1, y-1),\ (x-1, y+1),\ (x-1, y-1) \end{array} \right\}$.

- **Distancia Euclídea**

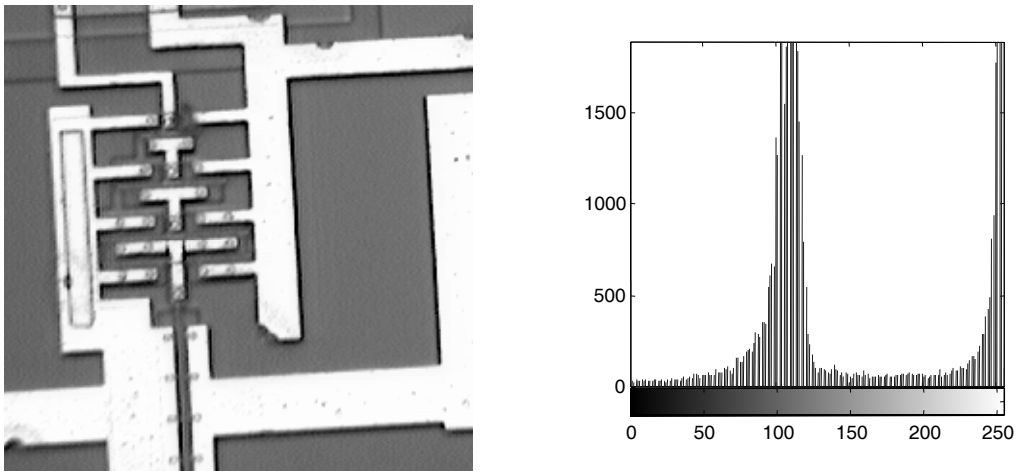
  Dados dos píxeles p y q de coordenadas (x, y) e (i, j) respectivamente, definimos la distancia euclídea como:

$$D_e = \sqrt{(x-i)^2 + (y-j)^2} \tag{1.1}$$

- **Histograma**

  El histograma es una distribución de frecuencias, es decir, expresa el número de veces que aparece cada uno de los posibles valores de la intensidad del nivel de gris en la imagen. El histograma de una imagen digital consiste en un diagrama de barras cuya abscisa representa los niveles de gris y cuya ordenada representa el número de pixel de la imagen para cada nivel de gris. Es frecuente normalizar el histograma entre 0 y 1 y emplear la frecuencia relativa de cada nivel de gris, es decir, el número de píxeles de un nivel dividido por el número total de píxeles de la imagen.

  Una imagen específica da lugar a un solo histograma. Sin embargo, un histograma puede provenir de imágenes diferentes. Es importante destacar que en el histograma se pierde toda la información espacial de la imagen ya que se conoce la distribución de los niveles de gris de la imagen pero se ignoran sus correspondientes coordenadas. En la figura1.3 (b) se representa el histograma obtenido para la imagen mostrada en (a).



(a) Imagen (b) Histograma

Figura 1.3: Ejemplo de histograma

- **Ruido**

  Se define cómo ruido a todo aquello que altera la imagen real de una escena. El ruido en una imagen provoca su degradación dificultando las operaciones que se puedan realizar sobre ella. Hay varias formas en las que el ruido puede afectar a una medida pudiendo ser clasificados como:

  - *Ruido blanco:* es un ruido ideal caracterizado por afectar por igual a todas las frecuencias.
  - *Ruido Gaussiano:* es una aproximación a una componente de ruido natural muy útil por sus propiedades matemáticas. La probabilidad de que un incremento (o decremento) x ocurra en un pixel cualquiera es:

    $$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{1.2}$$

    donde σ es la desviación típica de la variable aleatoria ruido y μ es su valor medio.
  - *Ruido "sal y pimienta":* significa que la imagen está corrompida en algunos píxeles con valores de intensidad muy diferentes a los de su vecindario, esto es, el valor que toma el píxel no tiene relación con el valor ideal sino

con el valor del ruido que pueden tomar valores muy altos o bajos. Se caracteriza entonces porque el píxel toma un valor máximo causado por una saturación del sensor o mínimo si se ha perdido su señal.

- *Frecuencial:* la imagen obtenida es la suma entre imagen ideal y otra señal, la interferencia, caracterizada por ser una senoide de frecuencia determinada.
- *Multiplicativo:* la imagen resultante es consecuencia de la multiplicación de dos señales.

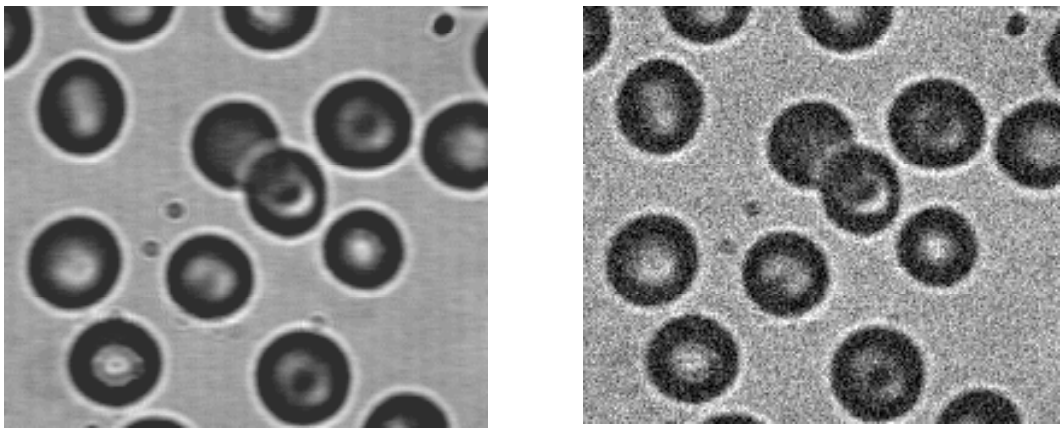
1.4 El problema de la visión.

La visión puede ser considerada como el sentido más poderoso pero a la vez el más complejo. En una primera opinión cabría pensar en que diseñar un sistema de visión es una tarea bastante sencilla dada la capacidad de cálculo que nos ofrecen los computadores actuales, capaces de resolver un sistema de ecuaciones diferenciales en escaso tiempo frente al tiempo requerido por un matemático. Sin embargo, esta idea es del todo errónea. A diferencia de los sistemas de ecuaciones diferenciales donde se conoce de manera precisa el conjunto de cálculos necesarios para su resolución, en el caso de la visión no somos conscientes de la complejidad del proceso ni de la secuencia de pasos que realizamos cuando analizamos una imagen siendo por tanto difícil reproducir nuestro comportamiento mediante un computador. Por este motivo el problema de la interpretación de imágenes por un sistema de visión artificial es un problema complejo estándo actualmente en fase de desarrollo. A este problema hay que añadir una serie de dificultades que complican aún más el proceso de interpretación de una imagen.

La primera dificultad que surje a la hora de interpretar una imagen reside en la propia naturaleza de la imágenes manejadas por el computador. En primer lugar éstas son digitales y además pueden estar afectadas por diferente ruido. En segundo

lugar, y no menos importante, en los niveles de intensidad de los píxeles intervienen un gran número de factores como son fundamentalmente: la iluminación de la escena durante la captura de la imagen, la geometría de los objetos, el color y la textura de las superficies y, por último los parámetros y distorsiones de la cámara empleada. Lo realmente complicado de esto es que la contribución individual de estos factores es difícil de precisar con lo cual se complica el proceso de interpretación, pudiendo provocar un mal reconocimiento de los objetos presentes en la imagen y en consecuencia realizar una mala interpretación.

En la figura1.4, se muestra un ejemplo de imagen corrompida con ruido gaussiano; en la figura1.5 se presenta un ejemplo en donde la iluminación ha afectado al nivel de gris de algunos de los píxeles apareciendo éstos como puntos blancos; y finalmente, en la figura1.6 se ha representado una imagen distorsionada tal y como se aprecia en la plantilla de puntos del fondo de ésta.



(a) Imagen (b) Imagen con ruido

Figura 1.4: Efecto del ruido Gaussiano en una imagen

Otra dificultad que también hay que tener en cuenta a la hora de diseñar un sistema de visión es que pueden existir infinitas escenas cuya proyección dan lugar a una misma imagen . Por tanto, dada una imagen es imposible determinar de forma unívoca la escena que a dado lugar a ella (ver figura1.7). Por ello para construir un sistema de visión artificial es necesario imponer restricciones físicas

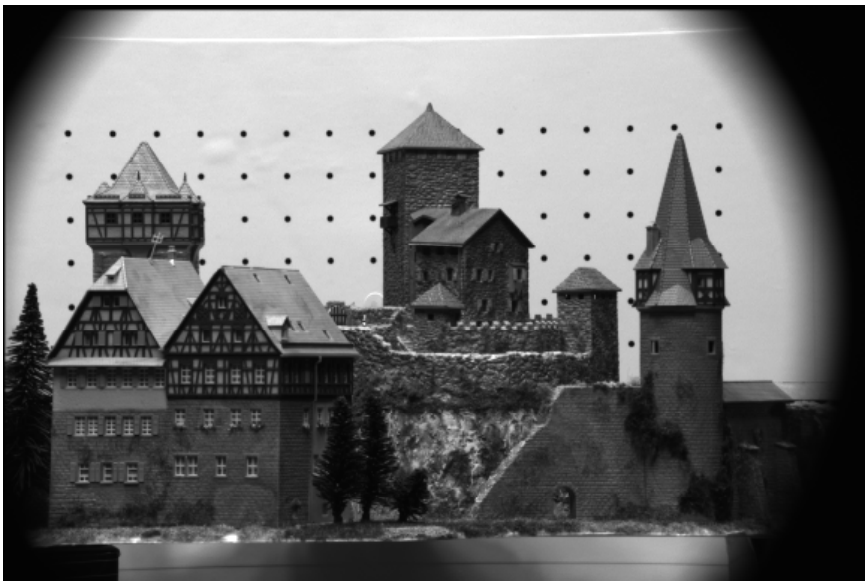
sobre los objetos del mundo real y sobre su proyección en imágenes. Y finalmente que cuando trabajamos con imágenes bidimensionales siempre va a existir una pérdida de información al pasar del mundo 3D a una imagen 2D.

Figura˜1.5: Efecto de la iluminación en los píxeles de la imagen

Veamos la magnitud del problema con un ejemplo concreto. Supongamos que miras la imagen mostrada en la figura 1.1 (a) y te preguntan que tipo de objeto sostiene la mano del robot. Tu podrías no haber visto nunca una mano de un robot y sin embargo eres capaz rápidamente de detectar los dedos del robot por la semejanza a tus propios dedos y por supuesto, averiguar el tipo de objeto que tiene en sus manos aproximándolo a uno de los modelos que existen en tu cerebro. Es decir, sin ninguna dificultad no sólo habrías detectado el objeto entre el resto de los elementos que aparecen en la imagen sino que también serías capaz de identificarlo como un cilindro. Por contra, este proceso no resulta nada sencillo si lo que disponemos es el conjunto de valores numéricos que representan a esa misma imagen es decir si lo que disponemos es la imagen digital (ver figura 1.1 (b)).

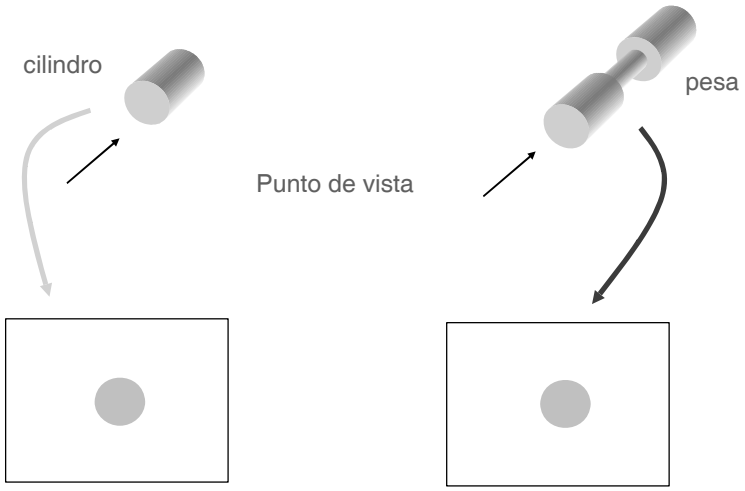
Por tanto, para poder llevar a cabo la tarea pedida con la información manejada por el compuador, es necesario estudiar los niveles de intensidad de cada pixel y comparar éste con sus vecinos. Así, por ejemplo si encontramos que existe un

cambio brusco de valor con alguno de sus vecinos es lógico pensar en que dicho pixel pertenece al contorno de uno de los objetos presentes en la imagen pero podría ser ruido existente en la imagen. Por todo ello el análisis de la imagen supone la realización de un conjunto de tareas relacionadas entre sí, que permitan alcanzar el objetivo marcado. Por supuesto, dependiendo de dicho objetivo será necesario realizar unos procedimientos u otros.



Figura˜1.6: Efecto de la distorsión introducida por algunas cámaras

Podemos decir que la diferencia fundamental entre nuestra forma de actuar y la del computador es que el ser humano dispone de una imagen analógica del mundo además de mucho conocimiento a priori para poder interpretarla mientras que el computador únicamente dispone, en principio, de una matriz de números. El computador será pues capaz de procesar partes locales de la imagen pero es muy difícil para él localizar conocimiento global. Para poder interpretar la imagen es esencial el conocimiento general, el conocimiento específico del dominio y la información extraída de la imagen.



Figura˜1.7: Infinitas escenas pueden dar lugar a una misma imagen

1.5 Etapas del procesado

Los modelos dominantes en visión artificial suponen la transformación de los datos sensoriales en descripciones significativas de la escena, mediante la utilización de etapas lógicas que emplean progresivamente representaciones más y más abstractas de la imagen original (ver figura 1.8). Aunque el estudio de dichas etapas será abordado en los siguientes capítulos, aquí se define el objetivo de cada una de ellas.

- **Adquisición**

  Es el proceso a través del cual se representan computacionalmente las magnitudes físicas externas, esto es, se plasma en una imagen digital el mundo captado por el sensor.

- **Preproceso**.

  Consiste en el tratamiento de la imagen digital para conseguir su mejora. Esta mejora suele consistir en: la eliminación de ruidos y/o el realce de la información que interese.

La entrada de la etapa de preproceso es una imagen con ruido $f(x, y)$ definida por las intensidades del nivel de gris de cada uno de sus píxeles y su salida es una imagen mejorada $g(x, y)$ definida también por las intensidades modificadas del nivel de gris de cada uno de sus píxeles.

Existen una gran variedad de operadores para mejorar la imagen. Algunos de ellos presentan ventajas en la eliminación del ruido, pero sin embargo deterioran ciertas características de la imagen. Otros en cambio, ayudan a resaltar las características de interés a cambio de introducir ruido. Por tanto, no existe un operador ideal que sea capaz de ofrecer una imagen mejorada libre de ruido y a la vez resaltar las características deseadas. El uso de uno u otro tipo de operador dependerá de la imagen que nos interese obtener para su tratamiento en las etapas posteriores involucradas en el proceso de visión artificial. En definitiva, depende íntegramente del tipo de problema que se pretende abordar.

- **Segmentación.**

  Es el proceso orientado a dividir la imagen digital en zonas disjuntas con significado propio.

  A pesar de invertir un esfuerzo considerable en la detección de los contornos que definen los objetos y las regiones, la segmentación sigue siendo muy difícil. En la detección de los contornos, por ejemplo, un umbral bajo permite detectar los bordes de bajo contraste pero también detecta los contornos falsos debido a las variaciones de las superficies; un umbral más alto es menos sensible tanto al ruido como a los verdaderos contornos. En el análisis de regiones, una región puede corresponder a más de una superficie, o por contra las variaciones de las superficies pueden dividir una superficie única en varias. Estos problemas pueden minimizarse estableciendo el contorno cuidadosamente y ajustando varios parámetros de umbral. Pero la obtención de elementos significativos no puede resolverse sin un conocimiento externo posiblemente heurístico. Un dibujo es perfecto cuando ha sido perfectamente interpretado pero una buena interpre-

tación no es posible sin un buen dibujo de líneas. Introducir el conocimiento del dominio es la forma de romper este círculo.

La cuestión está en hasta qué punto el proceso de detección de los distintos contornos debe ser dirigido por datos o por hipótesis. Es evidente que el conocimiento de los niveles más abstractos debe influir a los más primitivos. El uso de este conocimiento es muy complejo y necesita elaborados mecanismos para su control, dando lugar a implantaciones de sistemas muy especializados. Extender estos sistemas a dominios más amplios es difícil.

- **Descripción de los objetos**

  En la etapa dedicada a la descripción de los objetos podemos distinguir dos acciones: (1) Representación y (2) Extracción de las características o modelado. La representación nos permite encapsular en una estructura de datos o representación, manejada por el computador, la forma de los objetos resultantes de la segmentación de la imagen. La forma es una propiedad intrínseca de los objetos tridimensionales a partir de la cual pueden obtenerse otras propiedades tales como los contornos, normales a la superficie, etc. Dependiendo del tipo de información que sea posible extraer de la imagen o información de partida será conveniente el uso de una representación u otra. Una vez representados los objetos es necesario determinar el conjunto de características que permitan diferenciar dicho objeto de otros posibles objetos del dominio, es decir, el conjunto de características discriminantes del objeto (vector de características o modelo) que además sean invariantes a traslaciones, rotaciones y cambios de escala. En este problema se centra una parte de la visión por computador denominada usualmente modelado geométrico. El objetivo del modelado geométrico es la extracción de un conjunto de características invariantes ante rotaciones, traslaciones y homotecias que permitan reconocer el objeto extraído de la imagen a partir de una representación adecuada de su forma.

Veamos un ejemplo de modelo de un objeto. Supongamos que nuestro sistema de visión ha de ser capaz de reconocer entre monedas de cinco pesetas y monedas de veinte duros bajo el supuesto de que siempre las observa a la misma distancia y en la misma dirección. El modelo o vector de características que nos va a permitir posteriormente se reconocimiento podría ser simplemente su área (momento de orden cero). No obstante esta tarea no suele ser nada trivial, aún menos, cuando los objetos son tridimensionales en donde la apariencia del objeto depende del punto de vista en el que se haya adquirido la imagen.

- **Reconocimiento**.

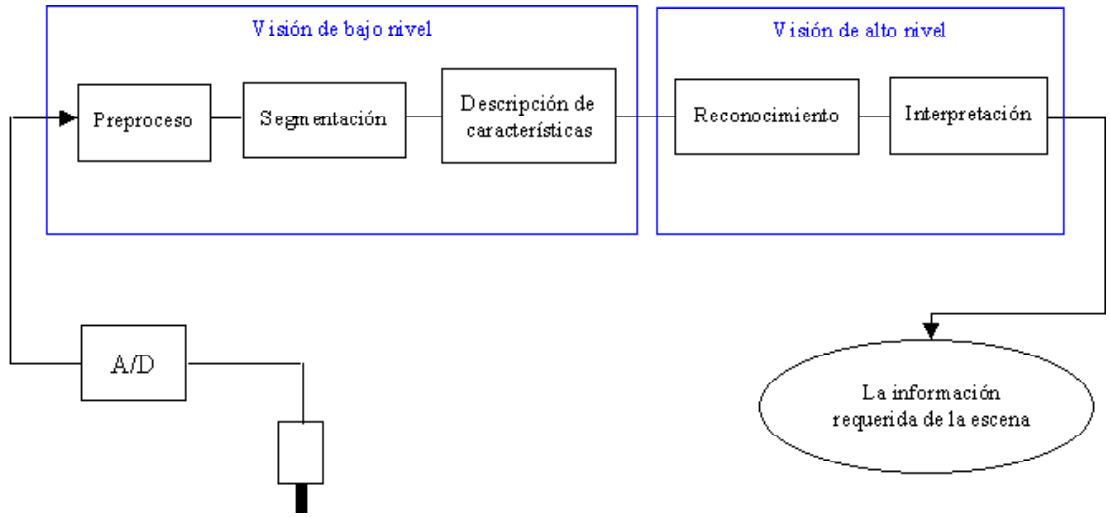
  Es el proceso gracias al cual asociamos a cada objeto, definido por un vector de características discriminatorias, un significado, esto es, un objeto del mundo real. No es más que encontrar una correspondencia entre el objeto presente en la imagen y la clase o prototipo cuyas características más se asemejan (matching).

- **Interpretación**.

  El objetivo principal de un sistema de interpretación de imágenes es construir una descripción simbólica de la escena que muestra la imagen. Para lograr este fin se utiliza la información obtenida en las etapas anteriores así como el conocimiento sobre restricciones y reglas que rigen el dominio de aplicación. Esta etapa está intimamente ligada con la IA estando, actualmente en una fase inicial sin resultados definitivos.

Es importante recordar que la división en las distintas etapas y niveles no implica independencia entre ellas, es más, muchos de estos procesos se relacionan unos con otros de muy diferentes formas. Por tanto, un sistema de visión debe entenderse como un conjunto de procesos que se interrelacionan.

Estas etapas pueden agruparse, a su vez, en dos niveles: visión de bajo nivel y visión de alto nivel. Las técnicas de visión de bajo nivel son equivalentes a las



Figura˜1.8: Etapas del proceso de visión

encontradas en el procesamiento digital de la imagen las cuales han sido utilizadas durante décadas. Estos métodos se dedican al procesamiento de la imagen con el fin de obtener propiedades locales a partir de las cuales se puedan determinar los atributos de la escena.

Una secuencia habitual de pasos en este nivel sería la siguiente (ver figura1.8). Una imagen capturada por un sensor y posteriormente digitalizada es manipulada por un computador con el propósito de eliminar el ruido y quizás resaltar alguna característica que pueda resultar interesante para la interpretación de la imagen. Un ejemplo de característica interesante podría ser la extracción de bordes. El siguiente paso es la segmentación de la imagen en la cual el computador intenta dividir la imagen en zonas u objetos. En este punto puede distinguirse entre segmentación total o parcial. La segmentación total sólo es posible en tareas muy simples (por ejemplo detectar un objeto negro sobre un fondo blanco). En problemas más complejos normalmente se realiza segmentación parcial en la cual únicamente se detecta en la imagen una determinada zona que cumple con ciertas características de color, de forma, etc. A continuación, se extraen las características de cada uno de los objetos presentes en la imagen, esto es, se genera un modelo de cada objeto el cual

permita durante la siguiente etapa reconocer al objeto y de esta manera interpretar la imagen.

La visión de bajo nivel normalmente utiliza muy poco conocimiento sobre el contenido de la imagen. Para dar al computador información sobre el contenido de la imagen es necesario la intervención de un experto humano que conozca el dominio de aplicación.

Los métodos más corrientes de procesado de bajo nivel fueron propuestos en los años 70. Las investigaciones más recientes en este área se centran en conseguir algoritmos más eficientes e implementarlos en equipos más sofisticados tecnologicamente (en particular en máquinas paralelas las cuales facilitan la enorme carga computacional de las operaciones realizadas sobre el conjunto de datos contenidos en la imagen).

Un problema complicado y todavía sin resolver es cómo ordenar los pasos que intervienen en el procesamiento de bajo nivel para resolver una tarea específica. Para cada caso es el operador humano quien define la secuencia de operaciones relevantes gracias a su conocimiento del dominio, resolviendo muchas causas de incertidumbre dependiendo, lógicamente, de la intuición del operador y de la experiencia previa. Por ello no es sorprendente que en los años 80 muchos proyectos se focalizaran en este sentido usando sistemas expertos. Será la visión de alto nivel la encargada de extraer y ordenar los pasos del procesamiento de la imagen empleando todo el conocimiento disponible. Por supuesto, este objetivo es extremadamente ambicioso.

La visión de alto nivel se basa en el conocimiento de objetivos y en la definición de planes que permitan lograr dichos objetivos. Intenta imitar el comportamiento humano y la habilidad para tomar decisiones dependiendo de la información contenida en la imagen. Por ello una forma de abordar el problema es empleando los métodos de la IA.

Se puede decir que la visión artificial se basa en el procesamiento a alto nivel y el proceso cognitivo se ajusta al conocimiento a priori sobre el contenido de la

imagen. Durante el procesamiento de alto nivel es necesario disponer de un modelo formal del mundo que sea comparado con la realidad percibida en la imagen digital. Cuando existan diferencias entre el modelo y la realidad percibida será necesario imponer nuevos objetivos parciales aplicados al procesamiento de bajo nivel con el fin de encontrar la información necesaria que permita actualizar la realidad percibida. De este modo se dipone de un bucle en el que los resultados obtenidos en el procesamiento de alto nivel definen tareas que han de realizarse en el procesamiento de bajo nivel llegando un momento en que dicho proceso converja a un objetivo global permitiendo interpretar la imagen.

Finalmente conviene destacar que el procesamiento de bajo nivel y el procesamiento de alto nivel difieren en los datos empleados. Así, la visión de bajo nivel utiliza como entradas a sus procesos una imagen digitalizada en forma de una matriz de datos, y genera como salida, en general, otra matriz de datos transformados. En cambio, en la visión de alto nivel se utilizan datos a nivel simbólico, es decir, representan elementos de un nivel de razonamiento simbólico, por ejemplo, áreas, formas, relaciones entre objetos.

1.6 Necesidad de conocimiento

El sistema de procesado descrito anteriormente representa un sistema de información lineal, es decir, las etapas se desarrollan secuencialmente, sin bucles y de manera independiente. La hipótesis de partida de este tipo de sistemas es muy fuerte, pues se asume que es posible identificar cada objeto de forma independiente, utilizando únicamente información local. Como veremos a continuación, éste no es el caso habitual. Es bastante probable que la salida obtenida en alguna de las etapas no sea la más adecuada lo que se traduce en la aparición de incertidumbre en la fase de interpretación. Podemos decir que son dos lo motivos que provocan la aparición de incertidumbre en el proceso: pérdida de información y errores cometidos durante el procesado.

Pérdida de información se produce ya en la propia captación de la imagen al pasar del mundo 3D a una imagen 2D, al representar sólo una parte del muno en un determinado instante de tiempo e incluso al introducir ruidos o distorsiones en la imagen que hacen que se pierdan o transformrn relaciones existentes en el muno real. Por otro lado, los términos utilizados para la interpretación no están contenidos en la propia imagen, sino en el modelo que representa el conocimiento del experto en el dominio (somos capaces de reconocer ua manzana porque disponemos previamente de un modelo asociado). Si los modelos son inexactos o imprecisos, porque no contengan todo el conocimiento, se introduce una pérdida de información al hacer computable el conocimiento. En cuanto a los errores de procesado pueden cometerse durante la captación de la imagen al realizar una calibración de la cámara imprecisa de forma que no se elimine completamente o correctamente la distorsión introducida por la cámara o durante la etapa de preproceso y segmentación cuando aplicamos una técnica inadecuada. En definitiva, teniendo en cuenta ambos factores, podemos encontrar incertidumbre en la aplicación de los operadores de procesado de bajo nivel, en la descripción del modelo y en la fase de interpretación al acoplar los objetos obtenidos con el modelo disponible.

Para conseguir buenos resultados sería necesario que el modelo, la imagen y los operadores fueran inequívocos. Aunque no es posible evitar la pérdida de información producida en el proceso de captación si que es posible luchar contra esta incertidumbre definiendo modelos precisos para lo cual será necesario definir modelos específicos para cada dominio de aplicación e introducir conocimiento para conseguir la aplicación correcta de los operadores.

De todo lo comentado podemos decir que el conocimiento permite reducir la incertidumbre del proceso y simplificar la transformación al reducir los espacios de entrada y/o salida de cada etapa.

A continuación vamos a describir los distintos tipos de conocimiento que pueden ser utilizados a lo largo del proceso. Encontramos diferentes tipos de conocimiento:

1. **Conocimiento del tipo de imagen**: tipo de imagen (RGB, HSI, escala de grises), estructura de píxeles (cuadrada, hexagonal, rectangular), etc.

2. **Conocimiento del proceso de formación de las imágenes**. Este conocimiento se utiliza en la calibración, eliminación de ruido, etc.

3. **Conocimiento de las relaciones existentes en el mundo**: conocimiento de propiedades físicas y geométricas. Por ejemplo, a no ser que tengamos evidencias de lo contrario, la línea que une dos puntos es una línea recta.

4. **Conocimiento específico del dominio de aplicación**: esto implica conocimiento de los objetos del dominio de aplicación, sus características y relaciones entre ellos. Es el conocimiento propio del experto (observador externo que realiza la interpretación). El conocimiento del dominio se compone de una parte declarativa y otra procedimental. La componente declarativa consiste en la descripción de los objetos de interés y las relaciones entre ellos. Los objetos de interés se describen en función de sus atributos: posición, forma, color, textura, tamaño, etc. Las relaciones entre objetos indican abstracción, composición, relaciones topológicas, relaciones espaciales (orientación, distancia, ...), etc. El conocimiento procedimental permite indicar cuándo y cómo es posible encontrar los objetos de interés en la imagen. Esto es, conocimiento de la estrategia más adecuada para la interpretación de la imagen. Existe un problema a la hora de que el experto exprese el conocimiento procedimental pues los procesos de visión de bajo nivel pertenecen al subconsciente y evidentemente es incapaz de definir el conjunto de pasos que realiza para determinar la región de un objeto en la imagen.

Dichos tipos de conocimientos pueden ser aplicados en la generación y confirmación de hipótesis tanto locales (etiquetado de regiones) como globales (interpretación de la imagen completa); para generar o confirmar hipótesis sobre el objeto del dominio al que corresponde cierta región de la imagen. Esto se puede hacer, por ejemplo,

localmente comparando las características de los objetos del dominio con las características de las regiones obtenidas en el proceso de segmentación; para confirmar o seleccionar hipótesis sobre la interpretación; o para establecer la estrategia de control global de inferencia. El uso de conocimiento durante el proceso puede ser explícito, cuando se indica por qué se selecciona un determinado operador e implícito cuando sólo el diseñador sabe por qué ha realizado tal selección. Esta forma de uso suele darse en las etapas de procesado de bajo nivel.

Finalmente, es importante comentar que cuanto más genérico sea el conocimiento utilizado mayor será el campo de aplicación del sistema de visión diseñado. Por tanto, interesa utilizar conocimiento genérico siempre que sea posible, dejando el conocimiento propio del dominio de aplicación para aquellas situaciones donde el conocimiento genérico no cumpla con las expectativas de interpretación o de computabilidad (tanto en el tiempo como en el espacio). En cualquier caso, en el nivel del conocimiento se debe expresar de forma explícita tanto el uso de conocimiento del dominio como del conocimiento genérico, pues esto aumenta la confianza en el sistema y facilita su mantenimiento y documentación.

1.7 Campos de aplicación

La utilización de la visión por computador está en continuo crecimiento introduciéndose cada vez mas en nuevas áreas así como consolidándose en las ya conquistadas. Entre los distintos campo de aplicación encontramos:

En la industria:

Los sistemas de visión en la industria permiten una mejor evaluación de magnitudes físicas así como buen desempeño de tareas rutinarias. Así vamos a encontrar distintas aplicaciones encaminadas a:

- Mejorar la calidad de los productos y de los procesos involucrados.

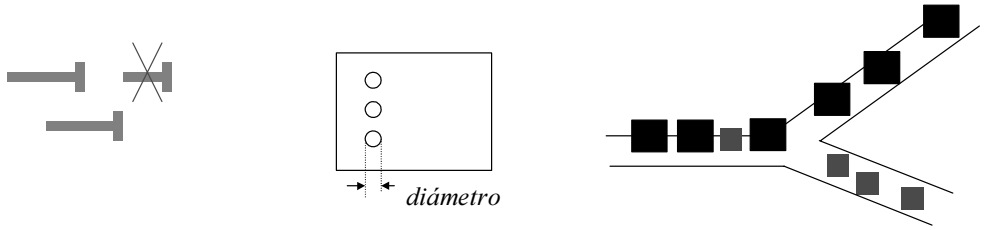
- Detectar pequeños defectos.
- Manipular las piezas con mayor precisión.
- Aumentar la rapidez de la inspección.
- Aumentar de la cadencia de producción.
- Evitar la presencia de operadores en entornos peligrosos (centrales nucleares).
- Abaratar costes de producción.

Sin embargo, no siempre es posible diseñar un sistema de visión debido a que éstos presentan ciertas limitaciones:

- Mala adaptación a situaciones imprevistas.
- Imposibilidad de ofrecer soluciones a distintas aplicaciones sin ayuda de personal cualificado.
- Dificultad de imitar la capacidad de reconocimiento humana.

Entre la aplicaciones que actualmente funcionan en la industria podemos citar (ver figura 1.9):

- Inspección visual automatizada o control de calidad de productos. Proceso automático para decidir si un determinado producto cumple con un conjunto de especificaciones previamente establecidas, definidas como estándar de calidad.
- Control de los procesos de automatización. Dicho control está íntimamente ligado a los elementos mecánicos o de control que intervienen en el proceso.
- Clasificación automática de piezas.
- Medición de objetos.



Figura˜1.9: Aplicaciones en la industria

- Aplicaciones compartidas:
  - Inspección de un producto a la vez que se corrige el proceso de fabricación.
  - Análisis de una pieza para el manipulado e inspección simultáneamente.

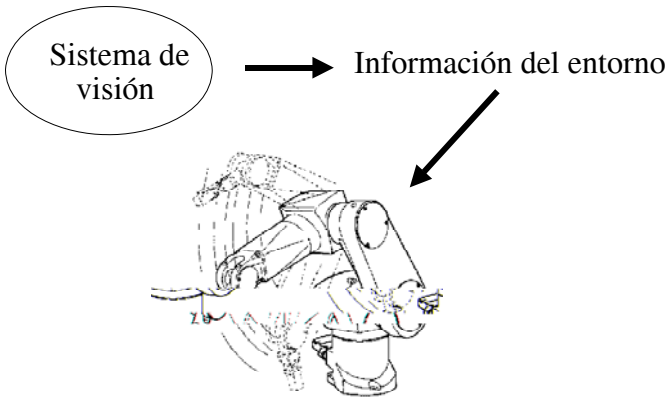
En robótica:

- Guiado de robots. El objetivo es conseguir sistemas robóticos capaces de ejercer cualquier trabajo de manera similar a como lo haría un ser humano para lo cual es necesaria la integración de los robots con algún tipo de sensor que suministre la información necesaria para generar la señal de control que los guíe (ver figura1.10) .

En medicina

- Pruebas de laboratorio automáticas. Un ejemplo lo encontramos en el recuento de globulos en la sangre o en la detección automática de células anómalas.
- Diagnóstico. Para la detección de determinadas zonas de la imagen las cuales pueden dar información sobre una patologia al experto médico.

En tele-medición



Figura˜1.10: Aplicaciones en robótica

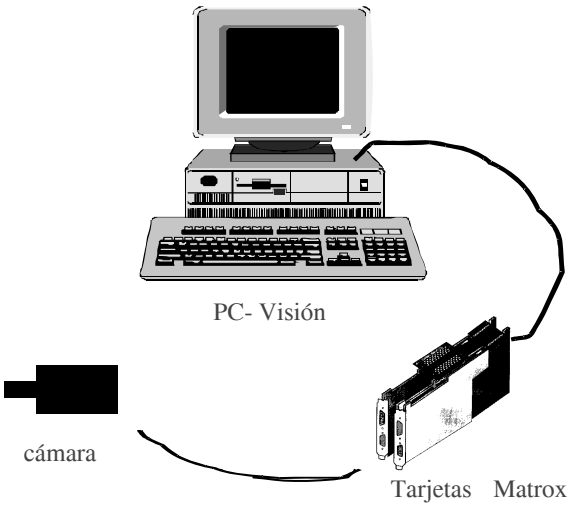
- Exploración geofísica. Un ejemplo lo encontramos en la interpretación de imágenes aéreas.
- Meteorología. Permite realizar un pronóstico meteorológico de las imágenes suministradas por los satélites.

Otros

- En entornos inaccesibles al ser humano tales como el espacio o el fondo submarino.
- Sistemas de seguridad. Un ejemplo sería la identificación de intrusos.

1.8 Componentes de un sistema de visión

Los componentes básicos de un sistema de visión son los que se muestran en la figura 1.11. La entrada del sistema de visión suelen ser una o varias cámaras de video, aunque también pueden ser cualquier otro tipo de sensor de imagen escáner óptico, escáner de resonancia magnético, sensor de tacto, sensor de rango, etc.



Figura˜1.11: Componentes básicos de un sistema de visión por computador

La información recogida por el sensor es enviada (normalmente de forma analógica) a un computador donde un hardware específico se encarga de la digitalización y, en algunos de los casos, del almacenamiento y procesamiento a bajo nivel de la imagen. Este hardware o tarjetas de visión pueden conectarse al computador mediante los buses más comunes , en particular, bus VME, bus PCI, bus ISA, etc. Las prestaciones de estas tarjetas varían enormemente y con ellas sus precios. Desde aquellas que tan sólo permiten la digitalización y el almacenamiento de una única imagen monocromo, hasta otras que incluyen el hardware para procesamiento en tiempo real de imágenes a color.

Para la visualización de imágenes capturadas puede ser necesario, dependiendo del tipo de tarjeta, un monitor analógico, además del monitor digital del computador. Finalmente, respecto al computador necesario puede ser cualquiera de los existentes en el mercado, condicionado, lógicamente al tipo de tarjeta de adquisición y a las prestaciones requeridas por el sistema que deseemos diseñar.

Si se hiciera una analogía entre un sistema de visión artificial y el sistema visual humano, los ojos serían la cámara o cámaras y el cerebro el computador. La función de la cámara es la de generar la señal de video a partir de la información luminosa

de la escena que será muestreada y digitalizada para su posterior tratamiento en un computador. La función del computador es la de manejar la imagen digitalizada con el fin de obtener la información requerida en cada aplicación.

Software

- **Scilab** (existe versión de Windows y Linux). Esta herramienta ofrece al programador un entorno cómodo para la simulación de algoritmos de visión.

  http://www-rocq.inria.fr/scilab

- **XITE 3.4.** Este software dispone herramientas que facilitan el procesado de imagen.

  http://www.ifi.uio.no/~blab/Software/Xite/WHAT_IS_XITE.html

- **Khoros** (versión estudiante válido para Linux). Khoros proporciona un entorno de programación visual, ofrece una variedad de herramientas para la manipulación de datos, procesamiento de señal, procesamiento de imágenes, análisis numérico, datos geométricos 3D, visualización de datos 3D y 2D, etc. En esta página también puedes encontrar un tutorial sobre procesamiento de imagen.

  http://www.khoral.com

Conocimientos nuevos adquiridos

El alumno debe ser consciente de las dificultades que entraña el análisis de una imagen y de que en los sistemas de visión por computador actuales el problema se resuelve mediante la aplicación de diferentes etapas de procesado las cuales transforman los datos sensoriales en descripciones significativas de la escena haciendo uso del conocimiento general así como del conocimiento específico del dominio de aplicación. Además debe comprender perfectamente términos tales como histograma, conectividad 4, etc.

Bibliografía complementaria

- Capítulo 1 de J. González: "Visión por computador". Paraninfo, 1999.
- Capítulo 1 de D. Maravall: "Reconocimiento de formas y visión artificial". Rama 1993.
- Capítulo 1 de A. De la Escalera: "Visión por computador. Fundamentos y métodos". Prentice Hall, 2001.
- Capítulo 1 de R. J. Schalkoff: "Digital Image Processing and Computer Vision". John Wiley & Sons, inc., 1989.
- Capítulo 1 y sección 2.3 de M. Sonka, V. Hlavac y R. Boyle: "Imagen Processing, Analysis and Machine Vision". Chapman & Hall Computing, 1993.

Actividades

En cuanto a la parte práctica se recomienda al alumno hacer las siguientes actividades mediante el empleo de cualquiera del software de visión disponible:

- Determinar el histograma de un conjunto de imágenes. Estudiar su aspecto en relación con los niveles de gris de la imagen. Por ejemplo: si la imagen es muy clara el histograma presentará la mayoría de los valores en la derecha y por el contrario, si es muy oscura la mayor densidad de valores estará en su zona izquierda.
- Corromper una imagen con diferentes tipos de ruido. Observar su aspecto. Esto ayudará en la etapa de preproceso a saber seleccionar adecuadamente el tipo de filtro que debe ser usado para eliminar el ruido.

Autoevaluación

Es conveniente que el alumno haya comprendido la naturaleza del problema así como las diferentes etapas que son posibles emplear para llegar a una solución

satisfactoria ante un determinado problema. Además debe ser capaz de a prever grosso modo el aspecto del histograma de una imagen y/o, si es posible, detectar el tipo de ruido con el que está corrompida.

Cuestiones de exámenes de los cursos anteriores.

Capítulo 2

Preproceso

Introducción y orientaciones para el estudio

En la etapa de preproceso realizamos el conjunto de tareas dedicadas a eliminar el ruido presente en la imagen y a realzar aquellas características de interés con el fin de mejorar la imagen captada por el sensor.

En este capítulo se van a describir las técnicas más empleadas en la etapa de preproceso. De manera más concreta, se estudiarán los siguientes puntos:

- **Transformaciones matemáticas**. Las transformaciones matemáticas son aquellos algoritmos aplicados sobre una imagen con el propósito de favorecer alguna característica presente en la imagen original o eliminar alguna otra que oculte lo que se está buscando.

- **Filtrado de imágenes**. Un filtro puede considerarse como un mecanismo de transformación de cierta señal de entrada dando lugar a una señal de salida diferente de aquélla. El objetivo del filtrado de imágenes digitales es restaurar o mejorar la imagen captada por el sensor. Desde este punto de vista se van a estudiar filtros que permitan eliminar el ruido y filtros que permitan realzar los bordes de los objetos.

- **Detectores de bordes**. Son las acciones dedicadas a detectar los bordes de los objetos. En una imagen continua, podría existir un borde cuando se

produce un cambio brusco de intensidad entre píxeles vecinos. Es decir, un borde se puede considerar como una discontinuidad en la función de intensidad de una imagen. Considerando esta definición un detector de bordes será un operador diferencial capaz de detectar las discontinuidades de una función. Estudiaremos dos técnicas: Técnicas de máximos locales y Técnicas de cruce por cero.

- **Transformaciones basadas en las intensidades del nivel de gris**. Una imagen puede analizarse considerando el conjunto de valores de las intensidades de sus píxeles como una simple distribución de intensidades. Las transformaciones basadas en el nivel de gris se basan en la manipulación del histograma a través de una función T que aplicada a cada una de las intensidades de la imagen original devuelve un nuevo valor de intensidad. El principal objetivo es conseguir histogramas más homogéneos aumentando de esta manera el contraste de la imagen. En esta parte se estudiarán diferentes tipos de funciones T.

Tanto para el alumno de Ingeniería de Sistemas como de Gestión este capítulo es completamente nuevo. Para su estudio es necesario que este habituado con la terminología usada en visión y con el histograma de la imagen teniendo muy clara la información que éste representa. Se recomienda durante el estudio que todas las técnicas de procesado descritas a lo largo de este tema, una vez comprendidas, sean probadas sobre diferentes imágenes utilizando para ello el software recomendado.

Objetivos

Conocer los procesos y tareas empleados en la etapa de preproceso.

2.1 Introducción

Tal y como ha sido comentado en el capítulo anterior, la imagen digital obtenida en la etapa de captación se caracteriza por la presencia de ruido, provocado por diferentes fuentes tales como las condiciones de iluminación, la propia adquisición de la imagen, el proceso de digitalización, e incluso la transmisión de la señal. El ruido se manifiesta normalmente como píxeles aislados que toman un valor de gris diferente al de sus vecinos. Por otro lado, en la mayoría de las ocasiones resulta muy interesante resaltar determinadas características presentes en la imagen para más tarde facilitar las etapas sucesivas que entran en juego en la visión artificial. Pues bien, es en la etapa de preproceso donde realizamos el conjunto de tareas dedicadas, por un lado, a eliminar el ruido presente en la imagen y, por otro, a realzar aquellas características de interés de manera que se transforme la imagen digital captada por el sensor visual, en una imagen mejorada caracterizada por reflejar las propiedades espaciales de la escena que sean de interés para la tarea.

Esta etapa de la visión está relacionada con otra disciplina como es el tratamiento de imágenes al compartir algunos de los algoritmos. Sin embargo, el enfoque es diferente ya que mientras el propósito en el tratamiento de imágenes es llevado por una persona en la visión artificial lo realiza íntegramente el ordenador.

Es importante hacer notar que la entrada de la etapa de preproceso es una imagen $f(x, y)$, posiblemente con ruido, definida por las intensidades del nivel de gris de cada uno de sus píxel y su salida es una imagen mejorada $g(x, y)$ definida también por las intensidades modificadas del nivel de gris de cada uno de sus píxeles (ver figura 2.1).

Existe una gran variedad de operadores para mejorar la imagen. Algunos de ellos presentan ventajas en la eliminación del ruido, pero sin embargo deterioran ciertas características de la imagen. Otros en cambio, ayudan a resaltar las características de interés a cambio de introducir ruido. Por tanto, no existe un operador ideal que sea capaz de ofrecer una imagen mejorada libre de ruido y a la vez resaltar

las características deseadas. El uso de uno u otro tipo de operador dependerá de la imagen que nos interese obtener para su tratamiento en las etapas posteriores involucradas en el proceso de visión artificial. En definitiva, depende íntegramente del tipo de problema que se pretende abordar.

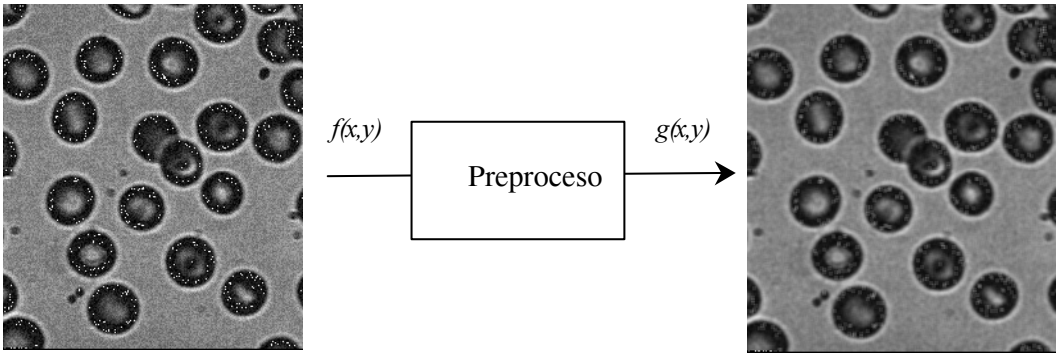


Figura 2.1: Diagrama esquemático de la etapa de preproceso.

En este tema se van a describir las técnicas más empleadas en la etapa del preproceso de la imagen. Es importante comentar que únicamente se van a utilizar imágenes estáticas.

La distribución de este capítulo es la siguiente: en primer lugar se describen las transformaciones matemáticas utilizadas en las diferentes técnicas aplicadas durante la etapa de preproceso. Seguidamente se exponen algunos de los métodos empleados para filtrar la imagen captada por el sensor tanto los dedicados a eliminar el ruido como los dedicados a resaltar alguna característica de interés presente en la imagen, en concreto los bordes de los objetos. Es importante aclarar que ambas acciones no son independientes, es decir, el mecanismo empleado para filtrar la imagen puede a su vez realzar alguna de las características deseada. En algunas ocasiones resulta interesante detectar los bordes de los objetos a partir de los cuales durante la fase de segmentación se dividirá la imagen en zonas disjuntas correspondientes, en principio, a los diferentes objetos que existen en la imagen. Es en la siguiente sección donde se estudian algunos de los algoritmos desarrollados para la detección de los bordes.

En la sección 5 se describen los métodos para la transformación de la imagen

basados en el histograma utilizados ampliamente en el tratamiento de la imagen con el propósito de mejorar su apariencia. Finalmente, en ocasiones, es necesaria la conversión de la imagen definida mediante la matriz de intensidades a una imagen binaria. Por ello, se dedica la última sección de este capítulo a describir algunos de los métodos existentes para llevar a cabo este proceso.

2.2 Transformaciones matemáticas

Las transformaciones matemáticas son aquellos algoritmos aplicados sobre una imagen con el fin de favorecer alguna característica presente en la imagen original o eliminar alguna otra que oculte lo que se este buscando. Dos son los tipos de transformaciones que pueden llevarse a cabo:

1. Transformaciones en el dominio del espacio: convolución y correlación.
2. Transformaciones en el dominio de la frecuencia: transformada de Fourier.

En las siguientes secciones se describe en detalle cada una de estas transformaciones.

2.2.1 Convolución

Se define la convolución bidimensional de la imagen *f(x,y)* respecto a la función *h(x,y)* a una nueva función *g(x,y)* tal que:

$$g(x,y) = f(x,y) * h(k,l) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} h(k,l) \cdot f(x+k, y+l) \qquad (2.1)$$

La carga computacional es elevada. Sin embargo, es posible realizar una simplificación en el filtrado de imágenes la cual nos permitirá disminuir dicha carga computacional. El método consiste en reducir el número de elementos de la función

$H(u,v)$ de manera que se mantengan sus propiedades de filtrado. Estas funciones impulsionales se denominan funciones de filtrado reducidas. Veamos un ejemplo:

Supongamos que la dimensión de la función impulsional es 3×3, esto es, el núcleo del filtro reducido es:

$$h(k,l) = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{2.2}$$

1. La imagen de salida al aplicar este núcleo viene dada por:

$$g(x,y) = \sum_{k=-1}^{+1} \sum_{l=-1}^{+1} h(k,l) \cdot f(x+k, y+l) \tag{2.3}$$

con $x, y = 0, 1, ..., N-1$. En este caso el número de operaciones necesarias se ha reducido considerablemente.

El núcleo reducido suele denominarse ventana o máscara lo cual está justificado por la propia naturaleza de la operación de convolución, que hace que este núcleo vaya barriendo píxel a píxel la imagen original. Así, para un píxel genérico $f(x,y)$ el correspondiente píxel de la imagen de salida se calcula mediante la expresión:

$$\begin{aligned} g(x,y) &= h_1 \cdot f(x-1,y-1) + h_2 \cdot f(x-1,y) + h_3 \cdot f(x-1,y+1) + \quad (2.4) \\ & h_4 \cdot f(x,y-1) + h_5 \cdot f(x,y) + h_6 \cdot f(x,y+1) + \quad (2.5) \\ & h_7 \cdot f(x+1,y-1) + h_8 \cdot f(x+1,y) + h_9 \cdot f(x+1,y+1) \end{aligned}$$

2.2.2 Correlación

La correlación de dos funciones reales continuas es una operación de similares características a la convolución con la salvedad de que no giraremos alrededor del origen los valores de una de las funciones. La expresión matemática para esta operación entre dos funciones discretas reales $f(x)$ y $h(x)$, viene dada por:

$$g(x) = f(x) \circ h(x) = \sum_{i=-\infty}^{i=\infty} \sum_{j=-\infty}^{j=\infty} f^*(i) h(x+i) \tag{2.6}$$

donde f^* es el complejo conjugado (en el caso de las imágenes, al ser números reales f^* es igual a f). Como hemos visto, para realizar la correlación, simplemente se desplaza $h(x)$ sobre $f(x)$ y se integra el producto desde -∞ hasta ∞ para cada valor de desplazamiento x.

La función de correlación suministra una medida de la similitud o interdependencia entre las funciones $f(x)$ y $h(x)$ en función del parámetro (el desplazamiento de una función con respecto a la otra). Si $f(x)$ y $h(x)$ son iguales, entonces la función de correlación se denomina función de autocorrelación.

En el caso bidimensional siguen siendo válidas expresiones similares. Así si $f(x, y)$ y $h(x, y)$ son funciones de variables discretas, su correlación se define como:

$$g(x) = f(x) \circ h(x) = \sum_{i=-\infty}^{i=\infty} \sum_{j=-\infty}^{j=\infty} f^*(i, j)h(x + i, y + j) \tag{2.7}$$

Una posible aplicación de la correlación es la localización de un patrón dentro de una imagen, ya que el valor donde la correlación es máximo corresponde a las coordenadas donde ese patrón se encuentra. Sin embargo si un objeto engloba a otro dará la misma respuesta. Otro caso en el que la correlación fallaría sería si buscamos un objeto grande y oscuro en una imagen donde existe también un objeto más pequeño y mucho más claro. Para solucionar esto se tiene la correlación normalizada que evita además posibles cambios en los niveles de gris entre el modelo que se busca y el objeto presente en la imagen.

2.2.3 Transformada de Fourier

La transformada de Fourier permite evaluar las propiedades frecuenciales o espectrales de una señal (continua o discreta) lo cual es muy útil para diseñar filtros que puedan modificar el comportamiento de la señal original de manera adecuada.

Se define la transformada de Fourier de una señal continua *x(t)* como la integral:

$$F[x(t)] = X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j \cdot 2 \cdot \pi \cdot f \cdot t} dt \tag{2.8}$$

Veamos la aplicación de la transformada de Fourier a la función periódica pura mostrada en la figura 2.2 y de ecuación:

$$x(t) = A \cdot \cos\ (2 \cdot \pi \cdot f_o \cdot t) \tag{2.9}$$

Su transformada de Fourier resulta:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j \cdot 2 \cdot \pi \cdot f \cdot t} dt = A \cdot \pi \cdot \delta(f - f_o) + A \cdot \pi \cdot \delta(f + f_o) \tag{2.10}$$

siendo δ la función Delta de Dirac. Durante el estudio solamente se tienen en cuenta las frecuencias positivas ya que son las únicas que tienen sentido físico.

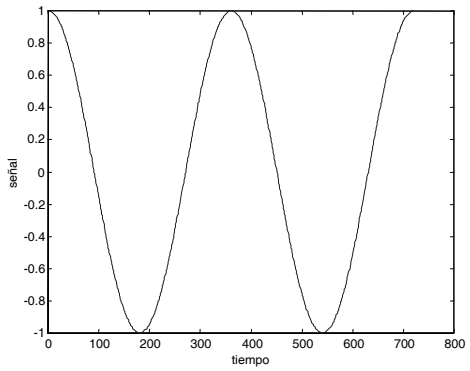


Figura 2.2: Función periódica pura.

Con este ejemplo sencillo se ha demostrado la utilidad de la transformada de Fourier ya que informa acerca del comportamiento de una señal en el dominio de la frecuencia. Aparentemente podría pensarse en su escasa utilidad ya que si se conoce la expresión matemática de la señal que se pretende evaluar es trivial obtener su comportamiento frecuencial. Sin embargo, en la mayoría de las situaciones, dicha expresión matemática es desconocida en cuyo caso lo más práctico es determinar la transformada de Fourier a partir de los valores de la función y su variable independiente y manejar directamente la información en frecuencia.

La versión discreta de la transformada de Fourier puede expresarse como:

$$X(u) = \frac{1}{N} \sum_{k=0}^{N-1} x(k) \cdot e^{-j \cdot \frac{2 \cdot \pi}{N} \cdot k \cdot u} \tag{2.11}$$

siendo u la frecuencia de la señal transformada con $u = 0, 1, ..., N-1$.

A continuación se definen dos conceptos importantes en relación con señales muestreadas:

1. Si T_s es el periodo de muestreo de una señal, entonces la resolución espectral o frecuencial de su transformada de Fourier es:

$$\Delta u = \frac{1}{N \cdot T_s} \tag{2.12}$$

   Esta relación nos dice que la resolución espectral será mayor cuanto mayor sea $N \cdot T_s$, esto es, el tiempo total de observación de la señal sin transformar.

2. El ancho de banda o frecuencia máxima de la señal transformada puede expresarse como:

$$u_{\max} = (N-1) \cdot \Delta u = \frac{N-1}{N \cdot T_s} \simeq \frac{1}{T_s} \tag{2.13}$$

   Esta expresión se conoce como teorema del muestreo de Nyquist que establece la frecuencia mínima de muestreo ($f_s = \frac{1}{T_s}$) que es preciso aplicar para poder recuperar todas sus componentes frecuenciales. En realidad, normalmente se considera un margen doble de seguridad en el muestreo. Es decir, que la frecuencia de muestreo f_s debe ser al menos el doble que la frecuencia máxima de interés.

$$f_s \geq 2 \cdot u_{\max} \tag{2.14}$$

Veamos un ejemplo. Supóngase una señal $x(t)$ como la mostrada en la figura 2.3 resultado de la superposición de dos señales periódicas puras: una señal de baja frecuencia y alta amplitud $x_1(t)$ y otra de frecuencia elevada y baja amplitud $x_2(t)$. Ésta puede expresarse como:

$$x(t) = x_1(t) + x_2(t) = A_1 \cdot \cos(2 \cdot \pi \cdot f_1 \cdot t) + A_2 \cdot \cos(2 \cdot \pi \cdot f_2 \cdot t) \tag{2.15}$$

donde $f_1 = 50Hz.$ y $f_2 = 1000Hz.$ La transformada de Fourier de $x(t)$ vendrá dada por:

$$V(f) = A_1 \cdot \pi \cdot \delta(f - f_1) + A_2 \cdot \pi \cdot \delta(f - f_2) \tag{2.16}$$

en donde se han ignorado las frecuencias negativas. Nótese cómo esta señal está compuesta por dos frecuencias puras f_1 y f_2. Si quisiéramos visualizar el espectro real de esta señal, se debe muestrear $x(t)$ con un periodo T_s de manera que se cumpla el teorema del muestreo:

$$T_s = \frac{1}{2 \cdot f_2} = 0'5\ ms \tag{2.17}$$

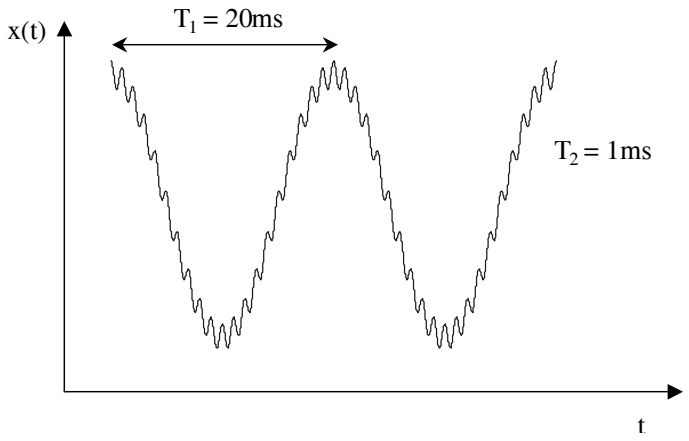


Figura 2.3: Señal constituida por dos frecuencias.

Es decir, sería necesario muestrear la señal dos veces cada milisegundo si se desea visualizar la frecuencia máxima. Por otro lado, supongamos que interesa calcular el espectro con una resolución mínima de 1Hz. Aplicando la ecuación (2.12):

$$\Delta u = \frac{1}{N \cdot T_s} = 1Hz \tag{2.18}$$

como $T_s = 0'5\ ms$, despejando N se obtiene que $N = 2000$. Es decir, habrá que tomar 2000 muestras de la señal $x(t)$.

Si se desea eliminar la componente de la señal $x(t)$ correspondiente a altas frecuencias se puede aplicar un filtro que la destruya. En este caso el filtro necesario es un filtro de paso bajo caracterizado por dejar pasar únicamente las frecuencias

bajas. Su función de transferencia frecuencial se muestra en la figura 2.4. Nótese que si f_c (frecuencia de corte del filtro) es menor a f_2 la señal $x_2(t)$ desaparecerá.

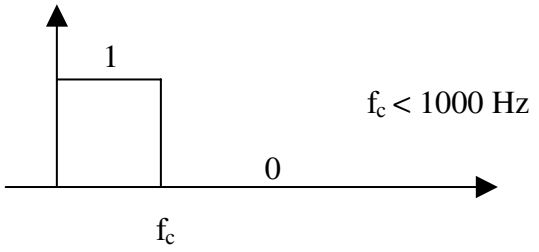


Figura 2.4: Función de transferencia funcional de un filtro de paso bajo.

Finalmente, para recuperar la señal original a partir de la transformada de Fourier basta con aplicar la expresión:

$$x(t) = F^{-1}[X(f)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(f)\cdot e^{j\cdot 2\cdot\pi\cdot f\cdot t}dt \tag{2.19}$$

o en su forma discreta:

$$X(k) = \sum_{n=0}^{N-1} X(u)\cdot e^{j\frac{2\cdot\pi}{N}\cdot k\cdot u} \qquad k = 1, 1, 2, ..., N-1 \tag{2.20}$$

Generalizando todas las ecuaciones a dos dimensiones que es el caso de interés en el presente contesto ya que las imágenes son bidimensionales se tiene:

$$f(u,v) = F[f(x,y)] = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cdot e^{-j\frac{2\cdot\pi}{N}\cdot x\cdot u}\cdot e^{-j\cdot\frac{2\cdot\pi}{N}\cdot y\cdot v} \tag{2.21}$$

$$f(x,y) = F^{-1}[f(u,v)] = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} f(u,v)\cdot e^{j\cdot\frac{2\cdot\pi}{N}\cdot u\cdot x}\cdot e^{j\cdot\frac{2\cdot\pi}{N}\cdot v\cdot y} \tag{2.22}$$

con u, v, x, $y = 0, 1, ..., N-1$.

2.3 Filtrado de imágenes digitales

Intuitivamente, un filtro puede considerarse como un mecanismo de cambio o transformación de una cierta señal de entrada, dando lugar a una señal de salida diferente

de aquélla. Para entender la utilidad de un filtro aplicado a una imagen digital es necesario, en primer lugar, entender el comportamiento y las propiedades frecuenciales de la imagen en cuestión.

Se entiende por frecuencia de una imagen digital la repetición de la imagen en una determinada dimensión. Como las imágenes digitales son bidimensionales únicamente existirán dos frecuencias, cada una de ellas correspondiente a una dimensión espacial. Pues bien, se dice que una imagen digital tiene un comportamiento de alta frecuencia cuando presenta grandes cambios de su intensidad luminosa en una zona espacial reducida. Análogamente, una imagen digital contiene bajas frecuencias cuando sus intensidades luminosas cambian suavemente.

El objetivo del filtrado de imágenes digitales es restaurar o mejorar la imagen captada por el sensor. La imagen $g(x, y)$ resultante de aplicar a otra imagen $f(x, y)$ un filtro con función impulsional $h(x, y)$ puede obtenerse de dos formas diferentes:

1. Utilizando su transformada de Fourier y su inversa según la siguiente secuencia de operaciones:

$$f(x,y) \xrightarrow{F} f(u,v); \qquad h(x,y) \xrightarrow{F} H(u,v)$$

$$g(u,v) = H(u,v) \cdot f(u,v)$$

$$g(u,v) \xrightarrow{F^{-1}} g(x,y)$$

2. Utilizando la convolución entre $f(x, y)$ y $h(k, l)$.

Dependiendo de si se pretende eliminar el ruido presente en la imagen o bien resaltar una determinada característica será necesario aplicar diferentes tipos de transformaciones. En los apartados siguientes se exponen algunos de los métodos utilizados para filtrar la imagen tanto cuando se quiere eliminar el ruido o cuando se pretende resaltar alguna característica como son los bordes de los objetos.

2.3.1 Eliminación del ruido

Tal y como se ha comentado anteriormente el ruido se manifiesta como píxeles aislados con un valor de gris diferente al de sus vecinos, es decir, le corresponde altas frecuencias. Los algoritmos de filtrado que se estudian seguidamente se basan en esta característica.

Operadores lineales

Dada una imagen $f(x, y)$, estos operadores determinan la intensidad asociada a un píxel (x, y) de la imagen mejorada $g(x, y)$ mediante la suma de la intensidades relativas a sus píxeles vecinos. Esto se realiza a través de la convolución de una ventana con las intensidades del nivel de gris de la imagen.

Filtro paso bajo

Un filtro paso bajo deja pasar únicamente las bajas frecuencias, o más exactamente, sólo aquellas frecuencias que se encuentren por debajo de un determinado umbral (frecuencia de corte). Este tipo de filtros se utilizan normalmente para la eliminación de ruidos y la reducción de los efectos producidos al emplear un bajo periodo de muestreo durante el proceso de digitalización de la imagen.

El ejemplo más simple es aquél que iguala el valor de la intensidad asociada a un píxel de la imagen mejorada $g(x, y)$, al valor medio de las intensidades de sus píxeles vecinos. Matemáticamente puede cuantificarse como:

$$g(x, y) = \frac{1}{M} \sum_{S} f(m, n) \tag{2.23}$$

donde S es la vecindad del píxel considerado y M el número de píxeles que forman parte de esa vecindad (considerando en ocasiones el propio píxel). Frecuentemente S es una vecindad rectangular (ventana) de (x, y), por ejemplo un cuadrado de $n \times n$. En este caso el operador se formula mediante la ecuación (2.24) en donde $h(k, l)$ es

la función impulsional cuyo valor en la región S viene dado por:

$$h(k,l) = \frac{1}{n^2} \tag{2.24}$$

Veamos un ejemplo práctico: supongamos que la imagen capturada por el sensor es la mostrada en la figura 2.5(a). Vamos a aplicar una función ventana de 3×3 expresada como:

$$\begin{matrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{matrix}$$

La imagen resultante $g(x,y)$ se obtiene mediante la convolución de la función ventana $h(k,l)$ con la función imagen inicial $f(x,y)$. En la figura 2.5(b) se muestra la imagen resultante tras el proceso de filtrado. Observamos cómo este operador deteriora los bordes de los objetos, presentando un aspecto más borroso en la imagen de salida. Por supuesto, dependiendo del tamaño de la ventana empleada se conseguirán diferentes resultados. Un ejemplo de esto se muestra en la figura 2.6 en donde se ha filtrado la imagen de la figura 2.5 (a) con un filtro de paso bajo utilizando una ventana de 5×5 y de 7×7.

Hay que comentar que este filtro presupone que la influencia de todos los píxeles es igual. Otra posibilidad es que cuanto más alejado esté el píxel del central su valor será menor. Se tiene entonces la ventana:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{matrix}$$

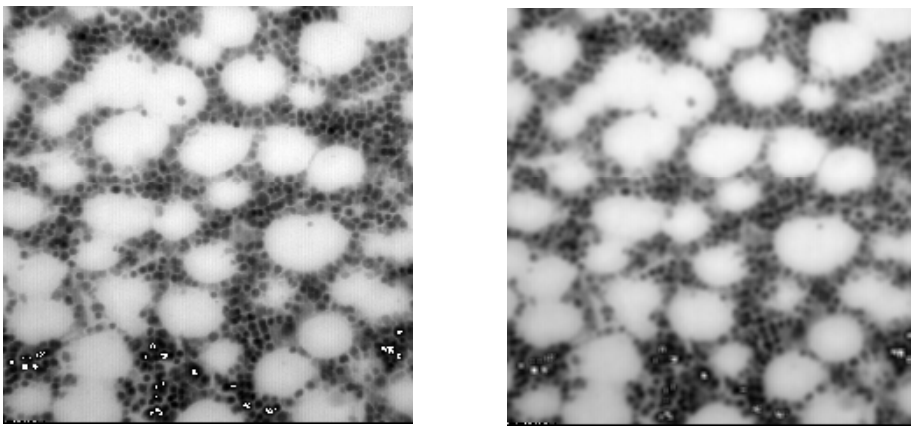
Si se quisiera dar más importancia al píxel central y a los vecinos tipo 4 se tendría:

$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

y de manera general:

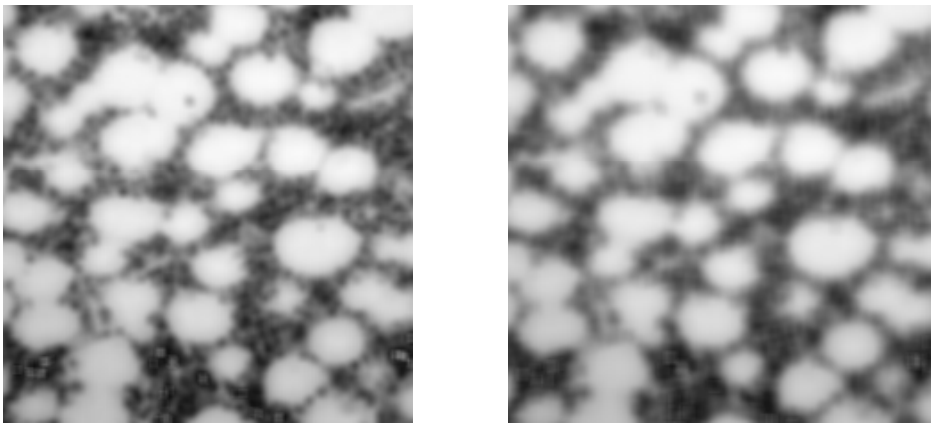
$$\begin{matrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{matrix}$$

debiendo ser la ganancia de todas ellas la unidad para no variar la imagen.



(a) Imagen (b)Ventana de 3x3

Figura 2.5: Imagen filtrada mediante un filtro paso bajo.



(a)Ventana de 5x5 (b)Ventana de 7x7

Figura 2.6: Imagen filtrada mediante un filtro paso bajo con diferente tamaño de la ventana.

Como se puede apreciar en los distintos ejemplos de estos filtros, mostrados en la figura 2.7 (c) y (d), presentan el inconveniente de que además de eliminar el ruido desdibujan los bordes de los objetos pudiendo eliminar incluso los objetos pequeños. Además, los filtros de paso bajo no eliminan el ruido impulsional.

(a) Imagen

(a) Filtro paso bajo

(c) ventana $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

(d) ventana $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Figura 2.7: Resultados obtenidos con otros tipos de filtro.

Gausiana

Otro tipo de máscaras son aquellas que imitan la forma de una gausiana:

$$G(x,y) = e^{-\frac{(x+y)^2}{2\sigma^2}} \tag{2.25}$$

Por ejemplo si $\sigma = 0'391$ la máscara resultante será:

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & 12 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

o bien si $\sigma = 0'625$ entonces:

$$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 7 & 11 & 7 & 2 \\ 3 & 11 & 17 & 11 & 3 \\ 2 & 7 & 11 & 7 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

Los filtros basados en gausianas tienen los mismos inconvenientes que los filtros paso bajo. En la figura 2.8 se muestra un ejemplo de aplicación de este tipo de filtros cuando $\sigma = 0'391$.

(a) Imagen

(b) Filtro Gausiana $\sigma = 0'391$

Figura 2.8: Resultado obtenido con un filtro Gaussiana.

Operadores no lineales

Los operadores lineales son simples y fáciles de implantar sin embargo, en ocasiones, no son suficientes para obtener los resultados deseados. Por ejemplo, el filtro de paso bajo reduce el ruido de la imagen pero provoca que los bordes de los objetos sean borrosos. Esto es un efecto indeseable. Por este motivo es importante estudiar operadores no lineales que cubran los mismos objetivos planteados para el caso de los operadores lineales pero eliminen sus desventajas.

Filtro Outlier

Cada píxel se compara con la media de sus ocho vecinos, si esta diferencia es superior a un valor preestablecido se considera ruido y se sustituye por el valor de

esa media. Por ejemplo, si la fila de píxeles fuera:

$$82 \quad 80 \quad 10 \quad 80 \quad 80 \quad 82$$

si se considera un entorno de 3 píxeles y el valor de umbral es 10, el resultado sería:

$$82 \quad 46 \quad 80 \quad 45 \quad 80 \quad 82$$

Aunque su respuesta ante el ruido impulsional es mejor que la de los filtros lineales ya que afecta a menos píxeles, no logra eliminarlo por completo al basarse en la media de los ocho vecinos siendo la media un operador en el que los valores extremos influyen mucho en el resultado.

Filtros de orden: filtro de la mediana

Dado un conjunto de N intensidades de píxel obtenidas para una región de la imagen local S y designadas por f_i, con $i = 1, 2, ...N$, definimos el orden:

$$R(\underline{x}) = \{f_1, f_2, ..., f_N\} \tag{2.26}$$

donde $f_i \leq f_{i+1}$. La intensidad asociada a un píxel de la imagen mejorada podría obtenerse mediante:

$$g(\underline{x}) = Rank_j \;\; R(\underline{x}) \tag{2.27}$$

siendo $Rank_j$ la intensidad asociada al elemento j de $R(\underline{x})$. Por ejemplo, si $j = 1$, se obtiene el filtro mínimo ya que se selecciona el primer elemento de $R(\underline{x})$ tratándose del píxel vecino de $\underline{x}$ con menor intensidad; esto es:

$$g(\underline{x}) = \min \;\; R(\underline{x}) = \min \;\; \{f(\underline{x}) \mid \underline{x} \in S\} \tag{2.28}$$

donde S es la vecindad elegida de $\underline{x}$. De manera análoga, el filtro máximo se obtiene para $j = N$ quedando:

$$g(\underline{x}) = \max \;\; R(\underline{x}) = \max \;\; \{f(\underline{x}) \mid \underline{x} \in S\} \tag{2.29}$$

El filtro de orden más utilizado es el conseguido para un valor impar de N seleccionando como intensidad asociada a un píxel de la imagen mejorada, la intensidad

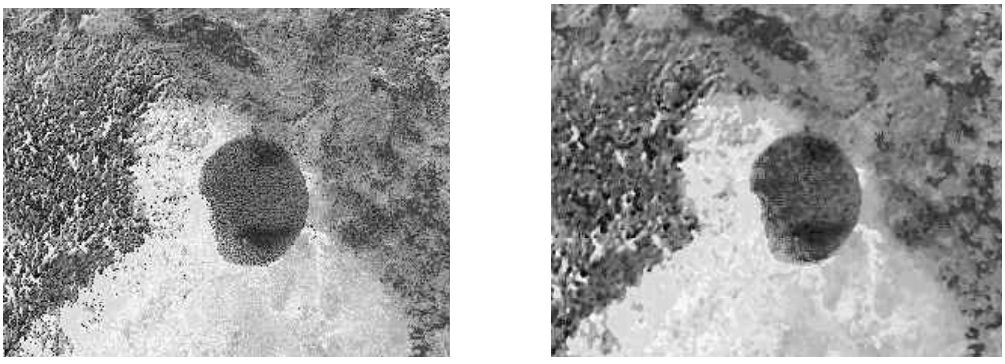
de la mediana f_m, es decir, para N muestras de intensidad se elige la posición:

$$m = \frac{(N+1)}{2} \tag{2.30}$$

Supongamos una imagen caracterizada por variaciones suaves en la intensidad del nivel de gris y por presentar ruido impulsional. Un filtro de paso bajo tendería a distribuir la intensidad del ruido en los píxeles situados alrededor del valor máximo o pico del ruido. En contra, un filtro de la mediana elimina este tipo de ruido en la imagen sin ningún tipo de degradación. El siguiente ejemplo ilustra esta idea.

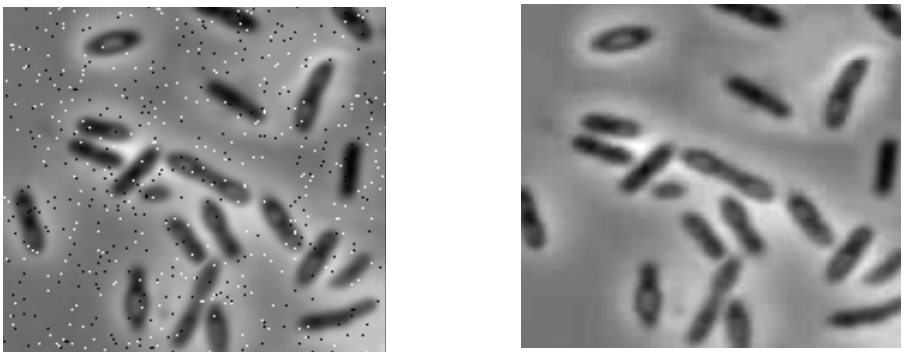
Consideremos que se eligen cinco muestras de intensidad en la imagen ($N = 5$) como vecindad del píxel situado en (x, y), esto produce $R(\underline{x}) = \{100, 110, 120, 130, 240\}$. Claramente, la intensidad del píxel de valor 240 es un ruido impulsional o una característica de la imagen. En este caso, la salida del filtro de la mediana es $g(\underline{x}) = 120$.

En las figuras 2.9 y 2.10 se muestran dos ejemplos de aplicación del filtro mediana. En el segundo de ellos, la imagen original fue corrompida con ruido sal y pimienta. Se observa como el filtro mediana es capaz de eliminarlo por completo sin prácticamente deteriorar la imagen.



(a) Imagen (b) Imagen filtrada con un filtro mediana

Figura 2.9: Imagen filtrada con el filtro mediana.



(a) Imagen (b) Imagen filtrada con un filtro mediana

Figura 2.10: Imagen filtrada con el filtro mediana.

La ventana seleccionada para el filtro de la mediana puede adquirir una variedad de formas tales como las mostradas en la figura 2.11. La forma de la ventana elegida afecta enormemente a los efectos del filtro. Así, por ejemplo, para conservar más los bordes verticales y horizontales de los objetos utilizaremos la ventana mostrada en la figura (b) mientras que si se quiere conservar los diagonales emplearíamos la representada en la figura (c). Además, la forma elegida para la ventana podría basarse en un conocimiento a priori de las características del ruido y de los objetos presentes en la imagen, tales como su orientación vertical u horizontal.

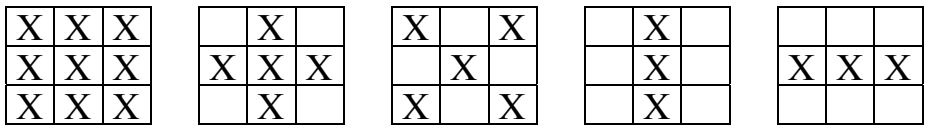


Figura 2.11: Diferentes formas de la ventana.

Para evitar cálculos extensos normalmente se utilizan ventanas pequeñas para implementar el filtro de la mediana. Además, la imagen podría ser filtrada repetidamente pasando múltiples veces esta ventana.

Se ha de resaltar que el operador de la mediana es un filtro no lineal ya que dadas

dos funciones imágenes producen dos secuencias $R_1(\underline{x})$ y $R_2(\underline{x})$, de manera que:

$$med\ (R_1(\underline{x}) + R_2(\underline{x})) <> med\ (R_1(\underline{x})) + med\ (R_2(\underline{x})) \tag{2.31}$$

Las propiedades que presenta este tipo de filtros son:

1. Reduce la varianza de la intensidades en la imagen. Así, el filtro de la mediana tiene la capacidad de alterar significativamente la textura de la imagen (ver figuras 2.9 y 2.10).

2. Dada una constante K y una secuencia $R(\underline{x})$, se cumple:

$$med\ (K \cdot R(\underline{x})) = K\ med\ R(\underline{x})$$

$$med\ (K + R(\underline{x})) = K + med\ (R(\underline{x}))$$

3. Suaviza las oscilaciones existentes en la intensidad con un periodo menor que el ancho de la ventana.

4. Cambia el valor medio de la intensidad de la imagen si la distribución de ruido espacial no es simétrica dentro de la ventana.

5. Conserva la forma de los bordes de los objetos además de su localización.

6. No se generan nuevos valores de gris es decir a un píxel de la imagen mejorada se le asigna un valor de intensidad existente en la imagen inicial.

7. La forma elegida para la ventana del filtro afecta a los resultados obtenidos.

La mediana tiene el inconveniente frente a los filtros lineales su lentitud. Así que el criterio general es que si el ruido es gausiano y la rapidez es un factor fundamental se usará la media o filtro paso bajo y en cambio si el ruido es impulsional o la rapidez no es un factor crítico la mediana.

2.3.2 Realce de bordes

Los operadores utilizados para realzar los bordes tienen un efecto opuesto a los empleados en la eliminación del ruido ya que de lo que se trata es de resaltar aquellos píxeles que presentan un valor de gris diferente al de sus vecinos. Por este motivo si la imagen está corrompida con ruido su efecto se verá incrementado por lo que resulta conveniente eliminar el ruido antes de aplicar un operador para realzar los bordes. Sin embargo, el filtrado de la imagen aparte de eliminar ruido también puede eliminar altas frecuencias que corresponden con los bordes de los objetos. Por consiguiente, cuando se aplican operadores para realzar los bordes sobre una imagen en la que previamente se ha eliminado el ruido, se han podido perder algunos de los mismos. Por ello debemos intentar llegar a un compromiso entre la capacidad de eliminar ruido y la posibilidad de resaltar los bordes de interés en la imagen estudiada.

Operadores lineales: Filtro de paso alto

El fundamento es el mismo, pero en esta ocasión sólo se dejan pasar las altas frecuencias, es decir, ayuda a enfatizar los bordes de los objetos. Una técnica caracterizada por resaltar los bordes genera la imagen de salida $g(x, y)$ a partir de la imagen de entrada $f(x, y)$ mediante:

$$g(x, y) = f(x, y) - f_{sm}(x, y) \tag{2.32}$$

donde $f_{sm}(x, y)$ es una versión suavizada de $f(x, y)$. Por ejemplo, $f_{sm}(x, y)$ puede definirse cómo el valor medio de las intensidades asociadas a sus píxeles vecinos sin incluir (x, y), es decir, puede expresarse como:

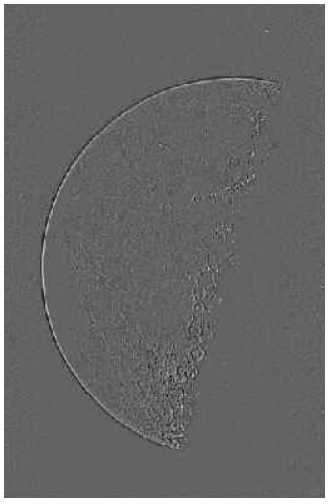
$$f_{sm}(x, y) = \frac{1}{8} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f(x + k, y + l) \qquad \text{para } k <> 0 \text{ y } l <> 0 \tag{2.33}$$

De este modo, la función ventana que implementa la ecuación (2.32) es:

$$\begin{matrix} \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} \\ \frac{-1}{8} & 1 & \frac{-1}{8} \\ \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} \end{matrix}$$

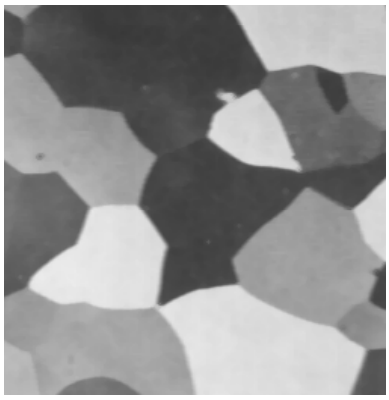
En las figuras 2.12 y 2.13 se muestra cómo quedaría una imagen cuando se filtra mediante un filtro de paso alto. Observamos algunos de los efectos indeseables cuando se utiliza un filtro de este tipo consistentes en la introducción de ruido y en la formación de bordes adicionales en la imagen de salida.

(a) Imagen original



(b) Imagen filtrada con un filtro paso alto

Figura 2.12: Imagen filtrada mediante un filtro paso alto.



(a) Imagen original

(b) Imagen filtrada con un filtro paso alto

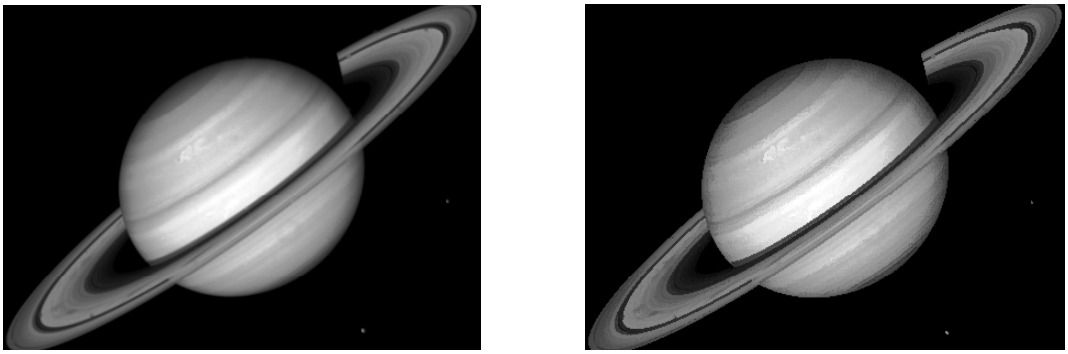
Figura 2.13: Imagen filtrada mediante un filtro paso alto.

Operadores no lineales: filtro max-min

El filtro max-min fue desarrollado por Kramer y Bruchner para el realce de los bordes. En el entorno de un píxel se obtienen los valores máximo y mínimo, siendo el nuevo valor:

$$g(x,y) = \begin{cases} f_{\max} & f_{\max} - f(x,y) \leq f(x,y) - f_{\min} \\ f_{\min} & \textit{si no se cumple} \end{cases}$$

En la figura 2.14 tenemos la imagen obtenida cuando se aplica un filtro max-min.



(a) Imagen original (b) Imagen filtrada con un filtro max-min

Figura 2.14: Imagen filtrada mediante un filtro max-min.

2.4 Detectores de bordes

El proceso de detección de bordes es una parte muy significativa de la etapa del preproceso dado que con ello se facilitan las etapas posteriores involucradas en visión artificial tales como la segmentación así como la detección de características.

La detección de bordes en una imagen no es una tarea sencilla. Lo que para una persona resulta una tarea realmente fácil y habitual, computacionalmente se convierte en una tarea complicada.

En primer lugar conviene definir de manera concisa lo que se entiende por borde. En principio un borde puede considerarse como la frontera que define a un objeto.

Esto supone el conocimiento a priori de lo que es un objeto, lo cual no es posible hasta que se conocen sus bordes. Luego, planteándolo de este modo, la detección de bordes es un problema recursivo que no tiene solución. Por tanto, es necesario definir el término "borde" de otra manera.

En una imagen continua, podría existir un borde cuando se produce un cambio brusco de intensidad entre píxeles vecinos. Así, probablemente, la onda mostrada en la figura 2.15 (a), (b) y (c) sería considerada como borde. Un caso interesante se presenta para una onda muestreada, figura 2.15(d), en donde cada par de píxeles con diferentes intensidades podría ser considerado como un posible borde. Resumiendo, un borde se puede considerar como una *discontinuidad* en la función de intensidad de una imagen. Ahora bien, una discontinuidad puede ser debida a muchas circunstancias propias de la dinámica de la escena, iluminación, ruidos, etc lo que puede provocar la detección de falsos bordes.

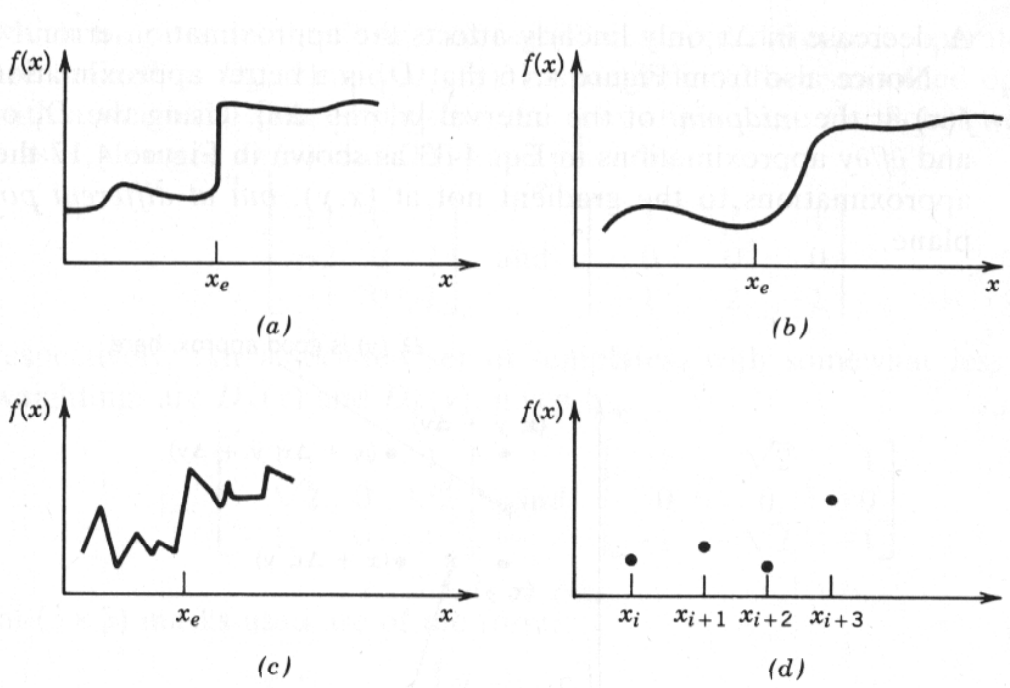


Figura 2.15: Ejemplos de bordes en la imagen. (a) Ideal (b) Aproximado. (c) Con ruido. (d) Muestreado. (R. J. Schalkoff, 1989).

Si partimos de esta segunda definición de borde, podemos ver que un detector de bordes será un *operador diferencial*, puesto que dichos operadores son capaces de detectar discontinuidades en una función.

La anterior afirmación se puede justificar desde dos dominios diferentes. En el dominio espacial, un borde es equivalente a una gran variación de intensidad en la dirección perpendicular al borde. Los operadores diferenciales son los que miden o detectan esas variaciones en una función. En el dominio de la frecuencia, un borde es una variación rápida de la señal (alta frecuencia) y un operador diferencial puede verse como un filtro de paso alto y por tanto filtrará las bajas frecuencias dejando en la señal de salida las altas frecuencias, es decir, los bordes.

Tal y como hemos dicho los operadores de realce de bordes se basan en operadores diferenciales. Atendiendo al operador diferencial utilizado podemos dividir las técnicas de detección de bordes en dos grandes grupos: de máximos locales y de cruce por cero. A continuación se exponen algunos de estos métodos.

2.4.1 Técnicas de máximos locales o aproximaciones basadas en la primera derivada

Cuando utilizamos un operador diferencial de primer orden los puntos de discontinuidad aparecen en los máximos de la función. Las derivadas parciales de primer orden de una función $f(x, y)$ se pueden aproximar de la forma:

$$\frac{\partial f}{\partial x} \cong D_1(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{2.34}$$

$$\frac{\partial f}{\partial y} \cong D_1(y) = \frac{f(y + \Delta y) - f(y)}{\Delta y} \tag{2.35}$$

Obsérvese que este operador corresponde a la convolución de la función imagen con el operador definido por:

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \text{ y } \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

para la coordenada X y la coordenada Y, respectivamente. Nótese que D_1 es la mejor aproximación de la pendiente en el punto medio del intervalo $[x, x + \Delta x]$. En este caso las aproximaciones se presentan en puntos diferentes al punto (x, y) (ver figura 2.16).

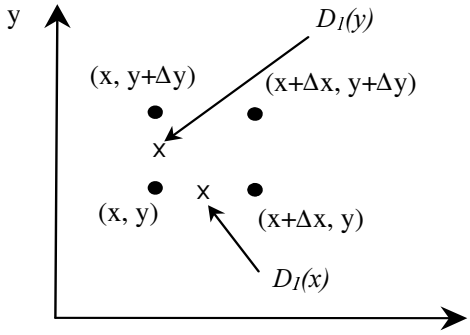


Figura 2.16: Aproximación derivada no centrada.

Estas máscaras no suelen utilizarse ya que son muy sensibles al ruido al tener en cuenta únicamente la información de dos píxeles.

Partiendo del operador D_1 surgen diferentes alternativas las cuales son menos sensibles al ruido:

Aproximación diferencial centrada

La primera alternativa al operador D_1 es la aproximación diferencial centrada, definida por:

$$D_2(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2(\Delta x)} \tag{2.36}$$

$$D_2(y) = \frac{f(y + \Delta y) - f(y - \Delta y)}{2(\Delta y)} \tag{2.37}$$

Los operadores $D_2(x)$ y $D_2(y)$ podrían implementarse mediante la convolución de la función imagen con las máscaras:

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \text{ y } \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

para la coordenada X e Y, respectivamente. Este operador es precisamente, el punto de partida para crear otras familias de máscaras para la detección de bordes:

1. Una variante de D_2 es el *operador de Prewitt* con máscaras de 3×3, $D_{2A}(x)$ y $D_{2A}(y)$, definidas como:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ y } \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

2. Otro operador con valores de peso que enfatizan el píxel central es el *operador de Sobel* con máscaras de 3×3, $D_S(x)$ y $D_S(y)$, definidas por:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ y } \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Mientras que el operador de Prewitt detecta mejor los bordes verticales el operador de Sobel lo hace en los diagonales.

Operador de Roberts

Otra alternativa muy empleada es el operador de Roberts definido por:

$$D_+\{(x,y)\} = f(x+\Delta x, y+\Delta y) - f(x,y) \tag{2.38}$$

$$D_-\{(x,y)\} = f(x, y+\Delta y) - f(x+\Delta x, y) \tag{2.39}$$

el cual puede implementarse mediante la convolución de la función imagen con las máscaras:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} y \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

La intensidad de salida en un borde de la imagen viene dada por:

$$f_{borde}(x,y) = \max\{|D_+|, |D_-|\} \tag{2.40}$$

Operador isotrópico u operador de Frei-Chen

Este operador intenta detectar tanto los bordes verticales como los diagonales. Las máscaras vendrán definidas en este caso como:

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} \; y \; \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

En la figura 2.17 se muestra la aplicación de estos operadores sobre la misma imagen. Quitando el operador de Roberts no se aprecian grandes diferencias en el resultado que proporcionan los otros tres operadores.

Combinación de operadores

Aunque los operadores anteriores podrían ser utilizados para obtener la información relativa a la dirección de los bordes, normalmente se emplean combinaciones de éstos para conseguir invariantes a la dirección del borde. Lo ideal sería conseguir un operador cuya salida no esté influenciada por la dirección de los bordes, es decir, disponer de un operador isotrópico.

Algunas de las combinaciones de gradientes más utilizadas son:

- $D_1(x)$, $D_2(x)$, $D_1(y)$, $D_2(y)$, D_+ o D_-: los cuales representan a los operadores individuales.
- $\sqrt{D_2(x)^2 + D_2(y)^2}$: favorece los bordes diagonales.
- $|D_2(x)| + |D_2(y)|$: favorece los bordes diagonales.
- $\sqrt{|D_+|^2 + |D_-|^2}$: favorece los bordes horizontales y verticales.
- $|D_+| + |D_-|$: favorece los bordes horizontales y verticales.
- $\max\{|D_+|, |D_-|\}$: operador isotrópico u operador de Roberts.

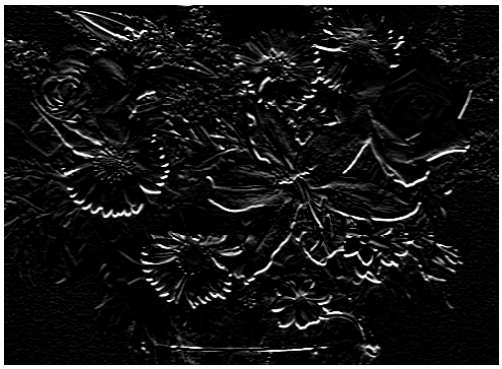
(a) Imagen original

(b) Operador de Roberts

(c) Filtro max-min en X

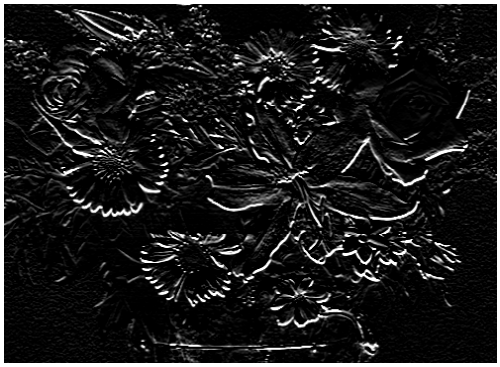
(d) Filtro max-min en Y

(e) Operador de Prewitt en X



(f) Operador de Prewitt en Y

(g) Operador de Sobel en X



(h) Operador de Sobel en Y

Figura 2.17: Resultados obtenidos con algunos de los detectores de borde estudiados.

En las figuras 2.19, 2.20 y 2.21 se muestran las imágenes obtenidas con los distintos operadores y diferentes combinaciones del gradiente aplicados a la imagen mostrada en la figura 2.18.

Figura 2.18: Imagen.

(a) $\sqrt{D_{2A}(x)^2 + D_{2A}(y)^2}$

(*b*) $|D_{2A}(x)| + |D_{2A}(y)|$

Figura 2.19: Resultado de la aplicación de combinaciones del operador de Prewitt.

(a) $\sqrt{D_S(x)^2 + D_S(y)^2}$ (b) $|D_S(x)| + |D_S(y)|$

Figura 2.20: Resultado de la aplicación de combinaciones del operador de Sobel.

(a) $\sqrt{|D_+|^2 + |D_-|^2}$ (b) $|D_+| + |D_-|$

Figura 2.21: Resultado de la aplicación de combinaciones del operador de Roberts.

2.4.2 Técnicas de cruce por cero

De la misma forma que se han empleado aproximaciones para máximos locales, también se pueden usar para otro tipo de operadores diferenciales de mayor orden. Así, cuando se utiliza la derivada segunda en una dirección, los bordes se encontrarán

en el *paso por cero* de la dirección dada. La figura 2.22(a) muestra un cambio brusco en el valor de la intensidad. En la figura 2.22 (b), (c) y (d) se muestra la derivada direccional segunda para varias orientaciones. Nótese el paso por cero en los tres casos.

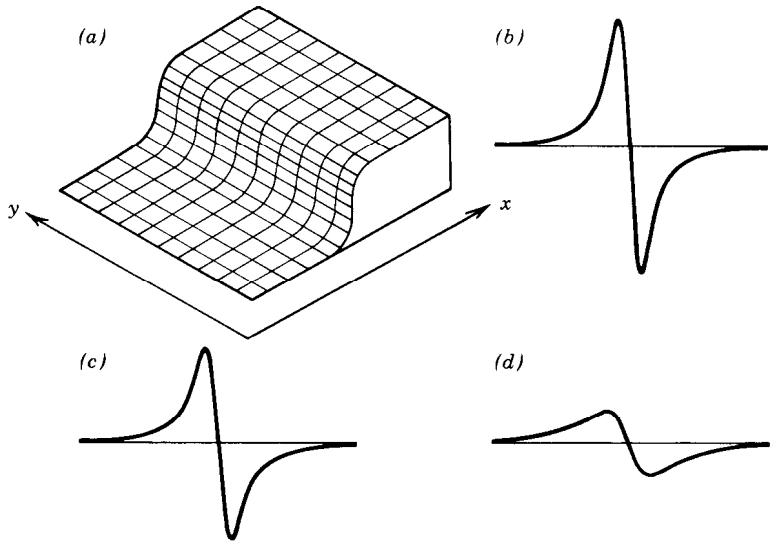


Figura 2.22: Cruze por cero de las derivadas direccionales segundas. (a) Muestra el cambio de intensidad. (b) dirección paralela al eje X. (c) 30 grados respecto al eje X. (d) 60 grados respecto al eje X. (R. J. Schalkoff, 1989).

Podemos establecer dos consecuencias claras:

(a) La derivada segunda cruza por cero para la localización correspondiente al punto medio del borde.

(b) Este efecto ocurre para todas las direcciones excepto para direcciones paralelas al borde. Esto nos permite determinar la orientación del borde.

Las aproximaciones de las derivadas parciales segundas en las direcciones X e Y son:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) \cong \Delta_x^2 = f(x+2,y) - 2f(x+1,y) + f(x,y) \qquad (2.41)$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right) \cong \Delta_y^2 = f(x, y+2) - 2f(x, y+1) + f(x, y) \quad (2.42)$$

Seguidamente se estudian dos tipos de operadores.

Operador Laplaciana.

Uno de los operadores diferenciales de realce de borde más utilizado es el operador Laplaciana, definido como:

$$\nabla^2 = \Delta_x^2 + \Delta_y^2 \quad (2.43)$$

cuya característica más interesante es que se trata de un operador anisótropo, no depende de la dirección, los pasos por cero detectarán los posibles bordes de la imagen.

El operador Laplaciana se puede expresar en forma matricial como (según el vecindario 4 u 8, respectivamente):

$$OLB_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}; \quad OLB_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

donde el píxel central toma el valor negativo de la suma de todos los que le rodean.

En la figura 2.23 se muestra el resultado de aplicar el operador de Laplaciana OLB_1 a la imagen mostrada en la figura 2.18. Como puede apreciarse este operador es muy sensible al ruido por lo que nunca se utiliza. Sin embargo si que es un paso intermedio para el operador Laplaciana de una Gausiana que se estudia a continuación.

Operador Laplaciana de una Gausiana.

Si elegimos el operador Laplaciana y lo aplicamos a una imagen previamente filtrada con un operador de suavizado de gausiana obtenemos el operador Laplaciana de la gausiana.

El operador gausiana se caracteriza por minimizar el efecto del ruido. Esta operación podría mejorar la conectividad entre bordes, Además, garantiza el cruce por cero de la segunda derivada. El operador de gausiana viene dado por:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \cdot e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2.44}$$

en donde el efecto del suavizado puede ser controlado mediante el parámetro σ. La imagen suavizada se obtiene a partir de la convolución de $G(x,y)$ con la imagen de entrada $f(x,y)$:

$$f_s(x,y) = f(x,y) * G(x,y) \tag{2.45}$$

Por otro lado, el operador Laplaciana permite detectar los bordes de los objetos. Aplicando dicho operador a la imagen suavizada obtenida a partir de la ecuación (2.45) se determina la imagen de salida. Esto es:

$$g(x,y) = (\bigtriangledown^2 G(x,y)) * f(x,y) \tag{2.46}$$

donde $\bigtriangledown^2 G(x,y)$ es el operador Laplaciana de una gausiana que puede expresarse como:

$$\bigtriangledown^2 G(x,y) = \left(\frac{1}{\pi\sigma^4}\right) \cdot \left(\frac{x^2+y^2}{2\sigma^2} - 1\right) \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.47}$$

En este operador la parte central negativa es un disco de radio $\sqrt{2}\sigma$. El dominio de la Laplaciana de la gausiana debe ser al menos del tamaño de un circulo de radio $3\sqrt{2}\sigma$.

En la figura 2.24 se puede ver el efecto del operador Laplaciana de una gausiana sobre la imagen mostrada en la figura 2.18 para diferentes valores del parámetro σ. Observamos que cuanto más estrecha sea la gausiana más bordes se obtendrán. Por tanto, este operador tiene el inconveniente de encontrar más bordes de los necesarios si la gausiana es estrecha y de redondear las esquinas.

Figura 2.23: Resultado de la aplicación del operador laplaciana.

(a)$\sigma = 0.5$ (b)$\sigma = 1$

Figura 2.24: Resultado de la aplicación de la Laplaciana de la Gaussiana para la detección de bordes

Detector de Canny

El detector de bordes de Canny se obtiene a partir de la optimización de una serie de condiciones:

- **Error:** Se deben detectar todos y sólo los bordes.

- **Localización:** La distancia entre el píxel detectado como borde y el real debe ser tan pequeña como se pueda.
- **Respuesta:** No se debe identificar varios píxeles como bordes cuando sólo exista uno.

Estas tres condiciones pueden ser expresadas de forma matemática como:

$$SNR = \frac{A\left|\int_{-w}^{0} f(x)dx\right|}{n_0\sqrt{\int_{-w}^{w} f^2(x)dx}} \tag{2.48}$$

$$Localización = \frac{A\left|f(0)\right|}{n_0\sqrt{\int_{-w}^{w} f^2(x)dx}} \tag{2.49}$$

$$Distancia = \pi\left(\frac{\int_{-\infty}^{\infty} f^2(x)dx}{\int_{-\infty}^{\infty} f'^2(x)dx}\right)^{\frac{1}{2}} \tag{2.50}$$

Canny busca la optimización del producto de la relación señal ruido por la localización y el tercero como una condición. Con ello se llega a que el operador óptimo es la derivada de una gausiana. Para obtener los bordes se siguen los siguientes pasos:

1. Se determina la gausiana unidimensional G de la imagen I.
2. Se obtienen las derivadas de la gausiana G_x y G_y.
3. Se convolucionan las derivadas de G con la imagen obteniendo I_x e I_y.
4. Se obtiene la magnitud M como la raíz cuadrada de la suma de los cuadrados de I_x e I_y.
5. Se eliminan aquellos píxeles que no sean máximos. Para ello se compara el valor de la magnitud para cada píxel con sus vecinos. Sólo aquellos que son máximos en su entorno se dejan como están.

6. Para detectar los bordes se utilizarán los umbrales T1 y T2, siendo este último el mayor. Si se usase el menor se detectarían bordes poco importantes, si se usase sólo el mayor se dejarían de detectar puntos pertenecientes a los bordes. Por ello la condición de pertenecer a un borde es ser mayor de T2 o mayor que T1 siempre que uno de sus vecinos sea mayor de T2.

En la figura 2.25 se muestra el efecto del operador Canny sobre la imagen mostrada en la figura 2.18.

Figura 2.25: Resultado de la aplicación del operador Canny.

2.5 Transformaciones basadas en las intensidades del nivel de gris

Una imagen puede ser analizada como si se tratara de un proceso estocástico, es decir, no se tiene en cuenta su información espacial (sus coordenadas) y, por tanto, se considera el conjunto de valores de las intensidades (niveles de gris) de los puntos como una simple distribución de intensidades en la cual cada valor de intensidad i tiene una probabilidad $p(i)$ de aparecer en algún punto de la imagen. Dicha función de densidad es precisamente el *histograma* de la imagen. Una ventaja evidente de las

transformaciones basadas en el histograma es la reducción en la carga computacional de la información existente en una imagen digital, ya que se pasa de una función bidimensional $f(x, y)$ con N^2 valores, a una función unidimensional con 2^q valores con q igual al número de bits empleados en la digitalización.

Siguiendo el convenio habitual, la escala de niveles de gris se establece de tal forma que los valores bajos corresponden a los niveles de intensidad luminosa oscuros y los valores altos a los niveles claros. Esto es, el negro le corresponde el valor inferior (0) y el blanco el valor superior (255 cuando se utilizan 8 bits para representar los niveles de gris).

Las transformaciones basadas en el nivel de grises se basan en la manipulación del histograma a través de una función T que aplicada a cada una de las intensidades de la imagen original, devuelve el nuevo valor de la intensidad (ver figura 2.26). Por tanto, la imagen transformada se obtiene al aplicar esta función a todos los puntos de la imagen original.

Un factor muy importante a tener en cuenta en las transformaciones de histograma es la utilización de intensidades del nivel de gris normalizadas entre 0 y 1. Esto es necesario para que se cumplan las propiedades de las transformaciones al considerar las imágenes como procesos estocásticos. Nótese que si tenemos un conjunto de L niveles de gris codificados con números entre 0 y $L-1$, se pueden conseguir valores entre 0 y 1, simplemente dividiéndolos por $L-1$.

Normalizando el nivel de intensidad en la escala de grises del histograma entre 0 y 1, las modificaciones del histograma se pueden visualizar como la siguiente función de entrada/salida:

1. Cuando $s = T(r) = r$ es decir, las intensidades de salida son iguales que las intensidades de entrada.

2. Cuando la pendiente de $T(r)$ es menor que 1 (por ejemplo para el punto a de la figura 2.27) la intensidad de los píxeles vecinos a a se oscurece ya que

el valor de las intensidades de salida es menor que el correspondiente valor de intensidad de la entrada. Además, se ha de resaltar que las diferencias existentes entre las intensidades de píxel (contrastes locales) se decrementan.

3. Cuando la pendiente es mayor que 1 (por ejemplo el punto b en la figura 2.27) se produce un efecto de iluminación de la imagen debido a que el valor de las intensidades de salida es mayor que el valor de la intensidades de entrada. Similarmente, en este caso, los contrastes locales se incrementan.

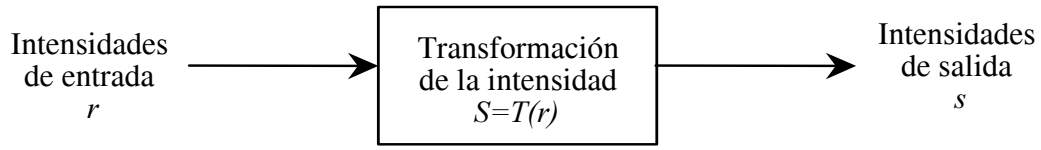


Figura 2.26: Transformaciones de la imagen modificando las intensidades.

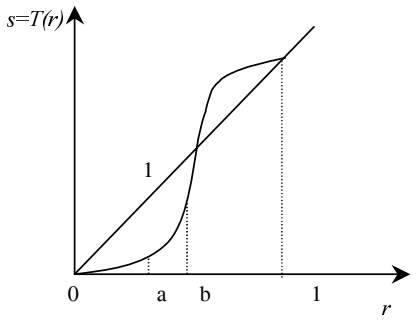


Figura 2.27: Función de entrada/salida.

Se pueden combinar simultáneamente estas funciones para producir diferentes efectos sobre la imagen original. Por ejemplo, si se aplica la función de entrada/salida mostrada en la figura 2.28 se está aumentando el contraste para los niveles de gris situados entre el valor de 0 y 0'2 expresados dentro de la escala normalizada y entre 0'4 y 1, manteniéndose sin modificar lo niveles entre 0'2 y 0'4. Los efectos de una determinada función sobre una imagen específica son imprevisibles, siendo obligada la experimentación controlada por un observador humano.

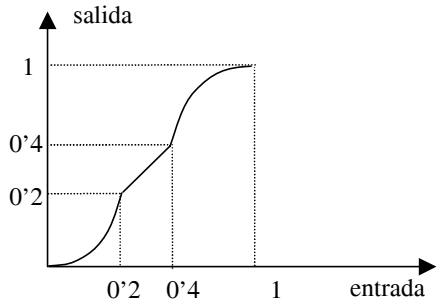


Figura 2.28: Función de entrada/salida.

Un caso particular es el mostrado en la figura 2.29 en donde $T(r)$ se define como:

$$T(r) = \{ \begin{array}{ll} A & \text{para } r > T \\ 0 & \text{en otro caso} \end{array} \tag{2.51}$$

Esta transformación es conocida como umbralización.

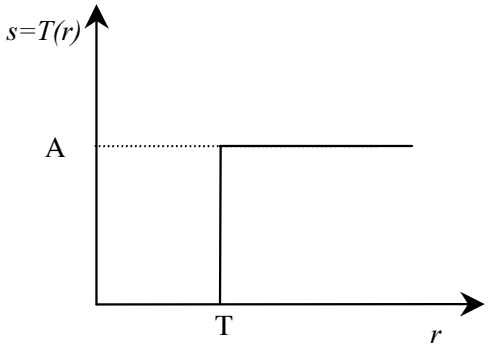


Figura 2.29: Transformación umbralización.

A continuación vamos a presentar algunas transformaciones típicas basadas en el histograma.

La primera transformación que vamos a estudiar es la función cuadrática:

$$g(x, y) = f^2(x, y) \tag{2.52}$$

siendo f y g los niveles de intensidad de la imagen digital de entrada y de salida, respectivamente. La función de entrada/salida correspondiente se representa en la figura 2.30 (a).

El efecto de esta transformación es el oscurecimiento, mejorando el contraste para los niveles de gris bajos (es decir, las intensidades oscuras), pero empeorando el contraste en los niveles altos.

Análogamente, la función cúbica, definida por:

$$g(x,y) = f^3(x,y) \tag{2.53}$$

acentúa aún más el efecto de oscurecimiento de la imagen original.

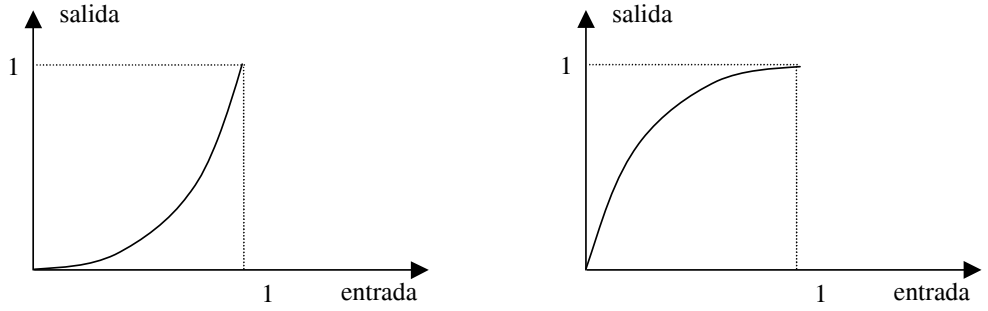
Otra transformación es la función raíz cuadrada, expresada por:

$$g(x,y) = [f(x,y)]^{\frac{1}{2}} \tag{2.54}$$

produce un efecto opuesto a la función cuadrática, esto es, la imagen de salida presenta un aspecto más blanco. Su función de entrada/salida se representa en la figura 2.30 (b). Similarmente, la función raíz cúbica:

$$g(x,y) = [f(x,y)]^{\frac{1}{3}} \tag{2.55}$$

acentúa todavía más el efecto de blanqueo.

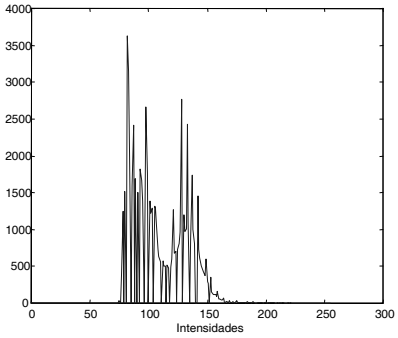


(a)Función cuadrática (b)Función raíz cuadrada

Figura 2.30: Representación de diferentes transformaciones.

En la figura 2.31 se representa el efecto de todas estas transformaciones sobre una misma imagen.

(a) Imagen



(b) Histograma

(c) Función cuadrática

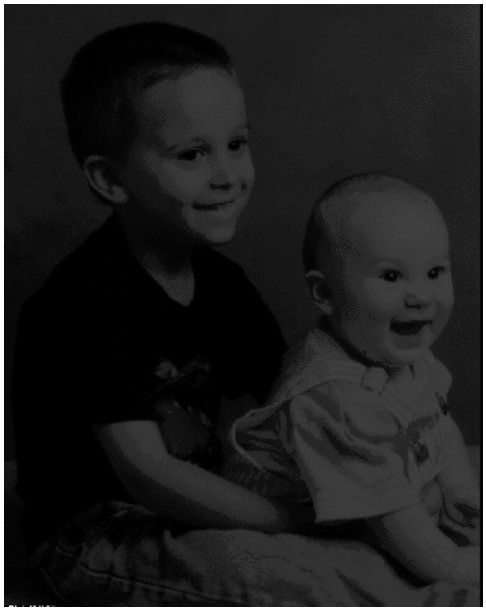
(b) Función cúbica

(c) Función raíz cuadrada

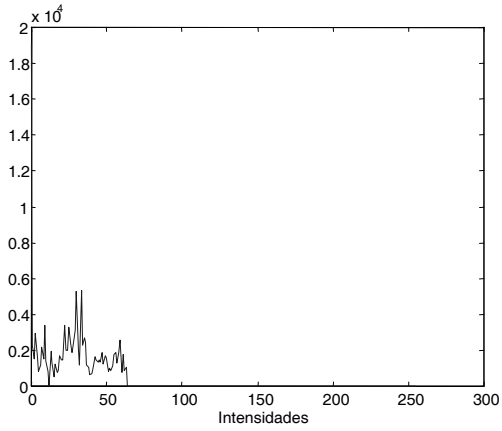
(b) Función raíz cúbica

Figura 2.31: Efecto de diferentes transformaciones sobre una imagen.

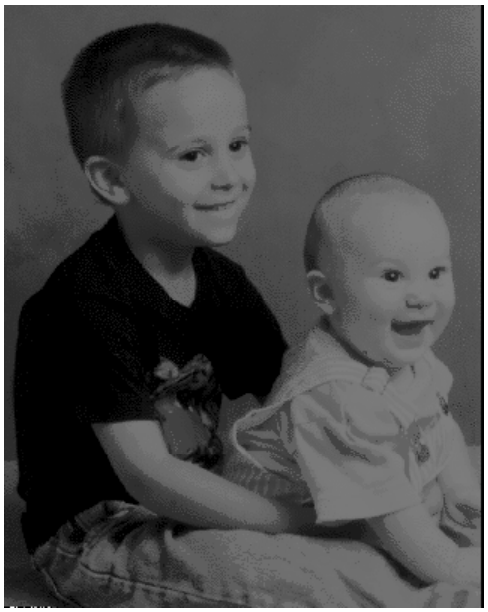
En las figuras 2.32 y 2.33 se observa la mejora conseguida en la imagen original después de aplicar la transformaciones más adecuadas.



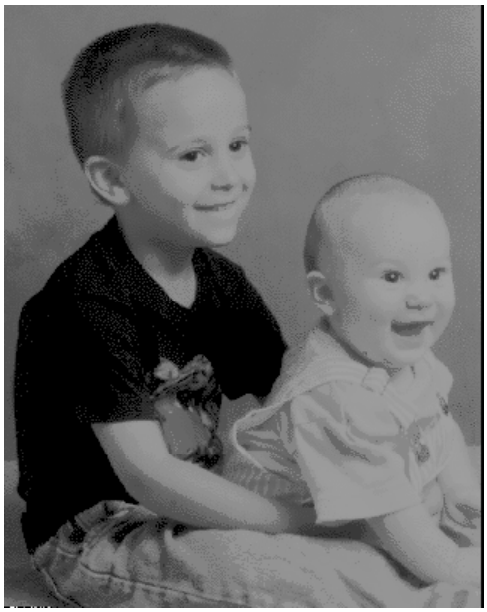
(a) Imagen



(b) Histograma



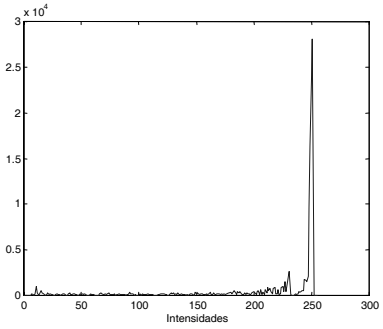
(c) Función raíz cuadrada



(b) Función raíz cúbica

Figura 2.32: Efecto de diferentes transformaciones sobre una imagen.

(a) Imagen



(b) Histograma

(c) Función cuadrática

(b) Función cúbica

Figura 2.33: Efecto de diferentes transformaciones sobre una imagen.

2.5.1 Modelo estocástico

Supongamos, sin pérdida de generalidad, que los valores de entrada se representan mediante una variable aleatoria r, la cual toma valores en el rango:

$$0 < r < 1$$

Además, se supone que los valores de salida $s = T(r)$ se incrementan progresivamente estando también comprendidos en el rango:

$$0 < T(r) < 1$$

y que existe su transformación inversa $T^{-1}(s)$.

Basándonos en la teoría de la probabilidad se demuestra que si $T(r)$ satisface la condición anterior entonces las funciones de densidad de probabilidad $p_r(r)$ y $p_s(s)$

de la variables aleatorias r y s respectivamente, se relacionan mediante la expresión:

$$p_s(s) = p_r(r)\frac{dr}{ds}|_{r=T^{-1}(s)} \tag{2.56}$$

De este modo, $T(r)$ proporciona una forma de modificar la función de probabilidad relativa a las intensidades del nivel de gris de la imagen de entrada. Encontramos dos métodos posibles basados en la manipulación del histograma: igualación del histograma y especificación del histograma.

Igualación del histograma

Analizando el histograma de una imagen es posible determinar sus características de contraste y de brillo. Por ejemplo, si $p_r(r)$ presenta valores distintos de cero para valores de r pequeños, implica que la imagen es oscura (ver figura 2.32). A la inversa, si $p_r(r)$ es distinto de cero para valores grandes de r equivaldría a una imagen luminosa (ver figura 2.33). En la figura 2.34 se muestra la interpretación del histograma. Estos efectos podrían ser causados bien por una iluminación inapropiada durante el proceso de captura de la imagen o bien por un nivel de sensibilidad del sensor inapropiado.

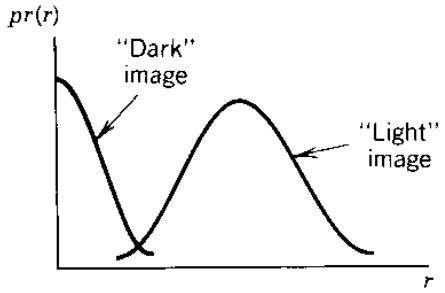


Figura 2.34: Interpretación del histograma.

Por supuesto, ninguno de estos casos es deseable puesto que el rango de intensidades es bastante estrecho haciendo que el contraste de la imagen percibida sea bastante pobre. La situación más deseable es disponer de una función de densidad uniforme ya que proporciona un contraste máximo.

Pues bien, el propósito de esta transformación es conseguir una nueva imagen con una distribución homogénea de intensidades, es decir, que todos los niveles de gris tengan una probabilidad similar. De esta forma se consigue un realce del contraste en los valores de la intensidad cercanos al máximo de la distribución y una disminución del contraste cerca del mínimo.

En principio, observando la ecuación (2.56) , parece lógico pensar que para una función de densidad de probabilidad arbitraria y dado que la situación más deseable es que $p_s(s) = 1$, una solución obvia podría ser:

$$\frac{dr}{ds} = [p_r(r)]^{-1} \tag{2.57}$$

Sin embargo, esta formulación no permite el tratamiento del caso en el que $p_r(r)$ sea igual a 0. Una alternativa considera cómo función de transformación del nivel de gris la expresión:

$$s = T(r) = \int_0^r p_r(a)da \tag{2.58}$$

la cual puede ser interpretada como una distribución de probabilidad acumulativa de la variable aleatoria r.

En resumen, el algoritmo descrito para la igualación del histograma consiste en la ejecución de dos pasos:

1. Calcular el histograma normalizado de los niveles de intensidad en la imagen de entrada $f(x, y)$, (esto es una aproximación de $p_r(r)$).

   Se entiende por histograma normalizado al histograma definido como una función de densidad de probabilidad de una variable aleatoria que varia entre 0 y 1. Esta variable corresponde a los niveles de gris normalizados entre 0 y 1.

2. Integrar esta aproximación de $p_r(r)$ utilizando la ecuación (2.58).

   Nótese que para imágenes digitales la función de probabilidad asociada al histograma es discreta, expresada por:

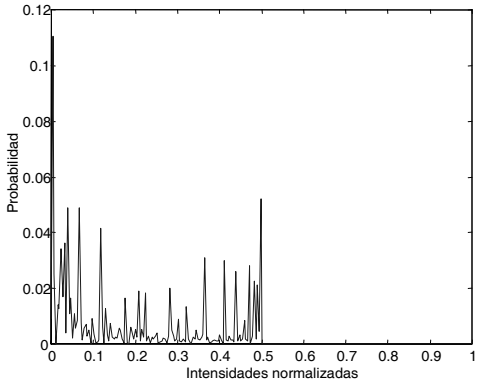
$$p_r(r_k) = \frac{n_k}{n} \tag{2.59}$$

con $0 \leq r_k \leq 1$ y $k = 0, 1, ..., 2^q - 1$ siendo q el número de bits del digitalizador, n_k el número de píxeles de la imagen digital con el nivel de intensidad normalizado r_k y n el número de píxeles total $(N.N)$. De este modo la función acumulativa discreta se puede escribir mediante la aproximación de la integral como:

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} \tag{2.60}$$

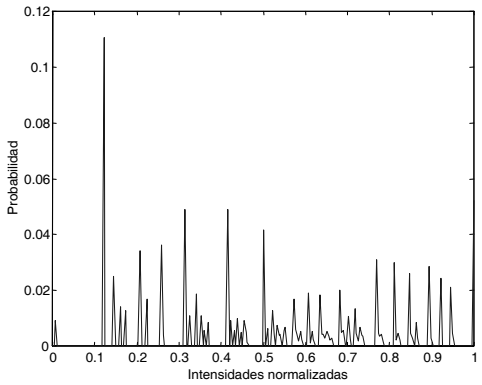
La figura 2.35 muestra los resultados obtenidos cuando se aplica una transformación basada en una igualación del histograma para el caso de una imagen con poco contraste.

(a) Imagen de entrada



(b) Histograma

(c) Imagen mejorada



(d) Histograma

Figura 2.35: Igualación del histograma.

Veamos un ejemplo sencillo del proceso de igualación del histograma. Supóngase

$\mathbf{r}_k$	$\mathbf{n}_k$	$\mathbf{p}_r(r_k) = \frac{n_k}{n}$
$r_0 = 0$	790	0'19
$r_1 = \frac{1}{7}$	1023	0'25
$r_2 = \frac{2}{7}$	850	0'21
$r_3 = \frac{3}{7}$	656	0'16
$r_4 = \frac{4}{7}$	329	0'08
$r_5 = \frac{5}{7}$	245	0'06
$r_6 = \frac{6}{7}$	122	0'03
$r_7 = \frac{7}{7}$	81	0'02

Tabla 2.1: Distribución de las intensidades de gris

una imagen de 64×64, con ocho niveles de intensidades de gris diferentes. En la tabla 2.1 se muestra la distribución de los niveles de gris.

La función transformación se obtiene utilizando la ecuación (2.60). Así:

$$s_0 = T(r_0) = \sum_{j=0}^{0} p_r(r_j) = p_r(r_0) = 0'19$$

$$s_1 = T(r_1) = \sum_{j=0}^{1} p_r(r_j) = p_r(r_0) + p_r(r_1) = 0'44$$

y,

$$s_2 = 0'65 \qquad s_5 = 0'95$$

$$s_3 = 0'81 \qquad s_6 = 0'98$$

$$s_4 = 0'89 \qquad s_7 = 1'00$$

Para poder representar los valores anteriores de las intensidades de gris de la imagen de salida es necesario transformarlos a los valores de las intensidades de gris más próximos entre los ocho niveles de gris posibles. Procediendo de este modo se obtiene finalmente:

$$s_0 = 0'19 \rightarrow \frac{1}{7} \qquad s_4 = 0'89 \rightarrow \frac{6}{7}$$

$$s_1 = 0'44 \rightarrow \frac{3}{7} \qquad s_5 = 0'95 \rightarrow 1$$

$$s_2 = 0'65 \rightarrow \frac{5}{7} \qquad s_6 = 0'98 \rightarrow 1$$

$$s_3 = 0'81 \rightarrow \frac{6}{7} \qquad s_7 = 1'00 \rightarrow 1$$

Especificación del histograma

Otro caso aparece cuando deseamos desarrollar una transformación de las intensidades de modo que la imagen de salida presente una función de densidad de probabilidad igual a una función de densidad de probabilidad específica. Una aplicación típica de este tipo de transformación se produce cuando se desea comparar dos imágenes de una misma escena tomadas por el mismo sensor con parámetros geométricos fijos, y captadas bajo condiciones de iluminación diferentes.

Dada una imagen de entrada con función de densidad de probabilidad $p_r(r)$, la transformación de los valores de la intensidad $s = T(r)$ se determina de manera que la imagen de salida presente una función de densidad de probabilidad igual a la deseada. El procedimiento es, en realidad, una versión modificada del caso anterior, consistiendo en los siguientes pasos:

1. Transformar las intensidades de la imagen de entrada para lograr una función de densidad de probabilidad uniforme. Las intensidades resultantes $s_1 = T_1(r)$ presentan una función de densidad de probabilidad uniforme para s_1(ver figura 2.36(a)).

2. Transformar las intensidades de la imagen de salida deseada para obtener una función de densidad de probabilidad uniforme. Resulta, así, una segunda transformación $s_2 = T_2(r)$ (ver figura 2.36(b)).

3. Invertir la transformación T_2 y combinar los resultados con la transformación T_1(ver figura (2.36(c)). Es decir, los valores de la intensidades de la imagen de salida vienen dados por:

$$s = T_2^{-1}(T_1(r)) \tag{2.61}$$

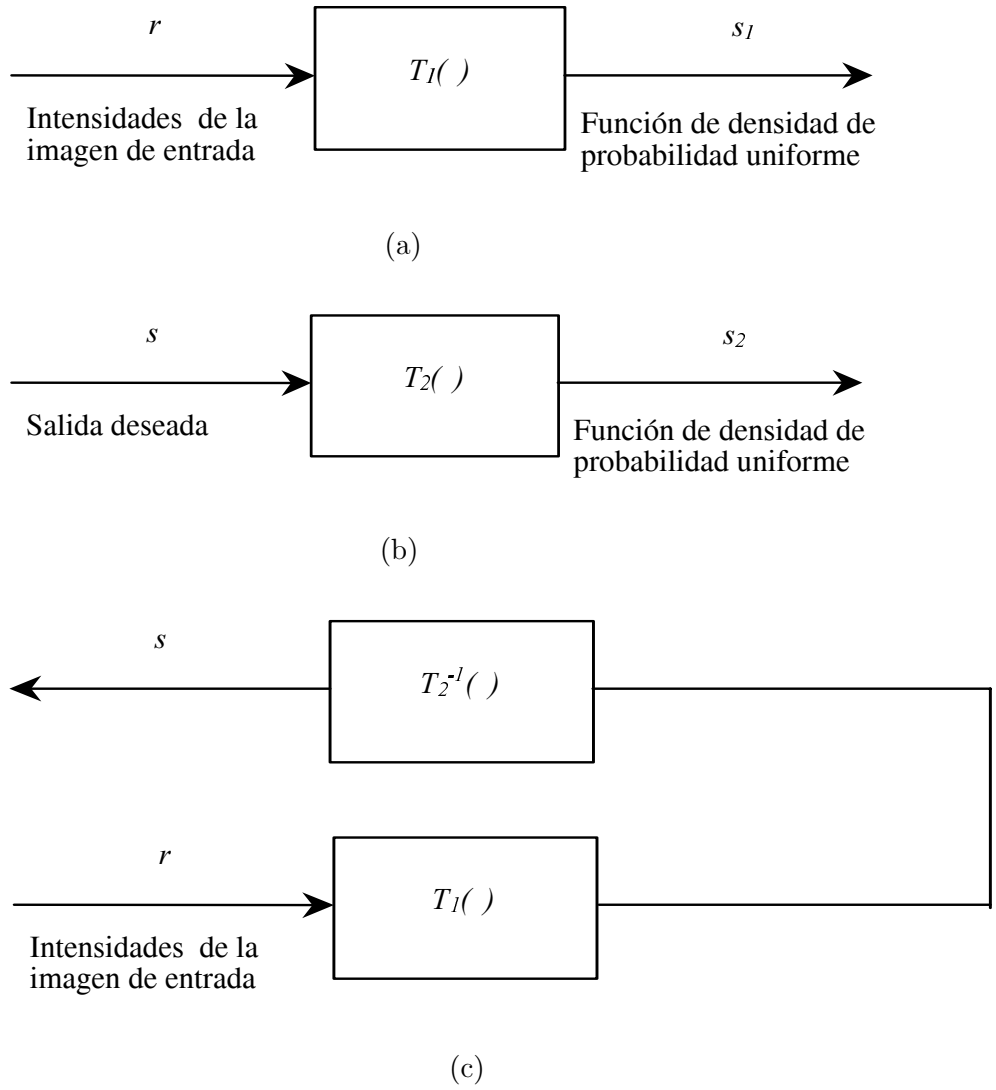


Figura 2.36: Proceso de especificación del histograma.

Software

Ver capítulo 2.

Conocimientos nuevos adquiridos

El alumno a lo largo del tema ha estudiado algunas de las técnicas que se aplican durante la etapa de preproceso. No sólo es necesario que haya aprendido cada una de estas técnicas sino que también debe saber cuando resultará más conveniente el empleo de una u otra y como puede influir en las etapas sucesivas.

Bibliografía complementaria

- Capítulos 4 y 5 de J. González: "Visión por computador". Paraninfo, 1999.
- Capítulos 10 y 11 de D. Maravall: "Reconocimiento de formas y visión artificial". Rama 1993.
- Capítulos 4 y 5 de A. De la Escalera: "Visión por computador. Fundamentos y métodos". Prentice Hall, 2001.
- Capítulos 3 y 4 de R. J. Schalkoff: "Digital Image Processing and Computer Vision". John Wiley & Sons, inc., 1989.
- Capítulo 4 de M. Sonka, V. Hlavac y R. Boyle: "Imagen Processing, Analysis and Machine Vision". Chapman & Hall Computing, 1993.

Además, algunos enlaces interesantes en este tema y temas posteriores son:

- http://www.css.tayloru.edu/~btoll/s99/424/res/model/geomod/geomod.html
- http://www-med.stanford.edu/school/vascular/RESEARCH/Image.htm
- http://www.ee.surrey.ac.uk/Research/VSSP/groups/shape/shape.html

Un buen tutorial que también propone actividades prácticas durante el estudio se puede encontrar en: http://www.khoral.com

Actividades

- Realizar y comprender los ejercicios resueltos suministrados por el equipo docente.
- Desarrollar utilizando cualquiera de las herramientas software los distintos algoritmos estudiados a lo largo de la asignatura. Ello permitirá afianzar conocimientos y detectar la dificultad del proceso en algunos casos. Para ello se suministrará al alumno una biblioteca de imágenes con las que podrá trabajar.
- Puede complementar sus conocimientos e incluso ampliarlos navegando por Intenet. En concreto se sugieren las siguientes direcciones donde puede encontrar diversos tutoriales sobre visión válidos para los temas de visión:
  - http://www.cm.cf.ac.uk/Dave/Vision_lecture/Vision_lecture_caller.html
  - http://www.dai.ed.ac.uk/HIPR2/
  - http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision0.html
  - http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip.html
  - http://www.dai.ed.ac.uk/CVonline/

Autoevaluación

Dada una imagen y los objetivos que se pretenden obtener, el alumno debe de saber si conviene aplicar algún método de preproceso para mejorarla o no y en su caso cúal.

También puede realizar las cuestiones y problemas de exámenes de los cursos anteriores.

Problemas resueltos

PROBLEMA 1

Dada la imagen representada en la figura mediante valores del nivel de gris de 0 a 255, calcular el valor del nivel de gris correspondiente solamente a los píxeles marcados en negrilla, que resulta cuando se le aplica:

1. Un filtro de la mediana
2. Un filtro de paso bajo.

Nota: Utilizar en ambos casos una ventana de 3×3.

10	9	9	100	105	102
11	10	**10**	**110**	100	105
8	9	**10**	**110**	105	103
150	160	**160**	**50**	52	54
153	159	**160**	**52**	52	55
150	151	153	52	53	50

¿Qué se pretende con este problema? En este problema se muestra un ejemplo de aplicación de dos de los métodos de filtrado estudiados. Además, con el segundo apartado se intenta mostrar al alumno un ejemplo de aplicación del proceso de convolución.

Solución

1. *Filtro de la mediana*

   La intensidad asociada a un píxel de la imagen resultante es la mediana de las intensidades de sus píxeles vecinos (incluido éste). De este modo, tomando los píxeles de izquierda a derecha y de arriba a bajo se tiene:

   $g(1,2)$: 9 9 9 10 **10** 10 100 110 110 $\Rightarrow$ 10

   $g(1,3)$: 9 10 10 100 **100** 105 105 110 110 $\Rightarrow$ 100

   $g(2,2)$: 9 10 10 10 **50** 110 110 160 160 $\Rightarrow$ 50

$g(2,3)$: 10 10 50 52 **100** 105 110 110 160 $\Rightarrow$ 100

$g(3,2)$: 9 10 50 52 **110** 159 160 160 160 $\Rightarrow$ 110

$g(3,3)$: 10 50 52 52 **52** 105 110 160 160 $\Rightarrow$ 52

$g(4,2)$: 50 52 52 151 **153** 159 160 160 160 $\Rightarrow$ 153

$g(4,3)$: 50 52 52 52 **52** 53 153 160 160 $\Rightarrow$ 52

La imagen resultante será:

10	9	9	100	105	102
11	10	**10**	**100**	100	105
8	9	**50**	**100**	105	103
150	160	**110**	**52**	52	54
153	159	**153**	**52**	52	55
150	151	153	52	53	50

2. *Filtro de paso bajo*

Aplicando la ecuación (2.4) y teniendo en cuenta que la ventana de 3×3 viene dada por:

$$\begin{matrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{matrix}$$

resulta para cada uno de los píxeles, analizados de izquierda a derecha y de arriba a bajo:

$$\begin{aligned}
g(1,2) &= \frac{1}{9}(9+9+9+10+10+10+100+110+110) = 41'8889 \\
g(1,3) &= \frac{1}{9}(9+10+10+100+100+105+105+110+110) = 73'2222 \\
g(2,2) &= \frac{1}{9}(9+10+10+10+50+110+110+160+160) = 69'8889 \\
g(2,3) &= \frac{1}{9}(10+10+50+52+100+105+110+110+160) = 78'5556 \\
g(3,2) &= \frac{1}{9}(9+10+50+52+110+159+160+160+160) = 96'6667 \\
g(3,3) &= \frac{1}{9}(10+50+52+52+52+105+110+160+160) = 83'4444 \\
g(4,2) &= \frac{1}{9}(50+52+52+151+153+159+160+160+160) = 121'8889 \\
g(4,3) &= \frac{1}{9}(50+52+52+52+53+153+160+160+52) = 87'1111
\end{aligned}$$

Como los valores de intensidad de gris se representan mediante enteros comprendidos entre 0 y 255, es necesario convertir los valores obtenidos anteriormente a enteros. Para ello aproximamos cada valor a su entero más próximo.

$$\begin{aligned} g(1,2) &= 41'8889 = 42 \\ g(1,3) &= 73'2222 = 73 \\ g(2,2) &= 69'8889 = 70 \\ g(2,3) &= 78'5556 = 79 \\ g(3,2) &= 96'6667 = 97 \\ g(3,3) &= 83'4444 = 83 \\ g(4,2) &= 121'8889 = 122 \\ g(4,3) &= 87'1111 = 87 \end{aligned}$$

La imagen resultante será:

1.

10	9	9	100	105	102
11	10	**42**	**73**	100	105
8	9	**70**	**79**	105	103
150	160	**97**	**83**	52	54
153	159	**122**	**87**	52	55
150	151	153	52	53	50

PROBLEMA 2

Implementar, utilizando pseudocódigo, el filtro de paso bajo. Considérese una ventana de 3×3 y una imagen de $N \times M$.

¿Qué se pretende con este problema? Con este problema se intenta introducir al alumno en la programación de las técnicas de filtrado. Este problema será el punto de partida para el estudio de los diferentes filtros aplicados sobre distintas imágenes, comprobando su efecto de una manera práctica.

Solución

N: número de filas en la imagen.

M: número de columnas en la imagen.

n: tamaño de la ventana.

$n \leftarrow 3$

$v \leftarrow \frac{1}{n^2}$

; Ventana para un filtro de paso bajo

$for\ i = 1\ hasta\ n$

$for\ j = 1\ hasta\ n$

$h(i,j) \leftarrow v$

end

end

; Convolución

$for\ x = 2\ hasta\ N-1$

$for\ y = 2\ hasta\ M-1$

$$g(x-1,y-1) \leftarrow h(1,1)\cdot f(x-1,y-1) + h(1,2)\cdot f(x-1,y)+$$
$$h(1,2)\cdot f(x-1,y+1) + h(2,1)\cdot f(x,y-1)+$$
$$h(2,2)\cdot f(x,y) + h(2,3)\cdot f(x,y+1)+$$
$$h(3,1)\cdot f(x+1,y-1) + h(3,2)\cdot f(x+1,y)+$$
$$h(3,3)\cdot f(x+1,y+1)$$

end

end

$Representar\ imagen\ g(x,y)$

fin

Obsérvese como ha sido tratado el problema que se plantean cuando realizamos

convoluciones en los píxeles pertenecientes al límite de la imagen dado que no existen los valores asociados a $f(x-1, y-1)$, $f(x-1, y)$, $f(x-1, y+1)$, $f(x, y-1)$ y $f(x+1, y-1)$. En concreto, se optó como solución no utilizar en la imagen de salida las dos columnas y las dos filas situadas en los extremos de la imagen, es decir, la fila 1 y la N y la columna 1 y la M. De este modo, la imagen de salida tendrá dos filas y dos columnas menos que la original. Evidentemente hacer esta manipulación, en principio, no supone ningún problema dado que la información que se pretende extraer de la imagen lógicamente no se encuentra en sus límites. Hay que comentar que existen otras soluciones igualmente válidas.

Se recomienda al alumno las siguientes actividades:

1. Implementar el algoritmo anterior relativo a un filtro de paso bajo en *scilab* probando su efecto sobre diferentes imágenes.

2. Modificar el algoritmo anterior para poder seleccionar diferentes tamaños de la ventana. Estudiar el efecto de la modificación del tamaño de la ventana sobre una misma imagen.

3. Realizar los pasos anteriores con otros tipos de filtro tanto lineales como no lineales.

PROBLEMA 3

Implementar, utilizando pseudocódigo, el *operador de Prewitt* para la detección de bordes. Considérese una ventana de 3×3 y una imagen de $N \times M$.

¿Qué se pretende con este problema? Con este problema se intenta introducir al alumno en la programación de los métodos para la detección de los bordes. Éste problema será el punto de partida para el estudio de los diferentes detectores de bordes analizados en la parte teórica del presente capítulo.

Solución

N: número de filas en la imagen.

M: número de columnas en la imagen.

n: tamaño de la ventana.

$n \leftarrow 3$

; Operador de Prewitt en la dirección x

$h_x(1,1) \leftarrow -1;\ h_x(1,2) \leftarrow 0;\ h_x(1,3) \leftarrow 1$

$h_x(2,1) \leftarrow -1;\ h_x(2,2) \leftarrow 0;\ h_x(2,3) \leftarrow 1$

$h_x(3,1) \leftarrow -1;\ h_x(3,2) \leftarrow 0;\ h_x(3,3) \leftarrow 1$

; Operador de Prewitt en la dirección y

$h_y(1,1) \leftarrow 1;\ h_y(1,2) \leftarrow 1;\ h_y(1,3) \leftarrow 1$

$h_y(2,1) \leftarrow 0;\ h_y(2,2) \leftarrow 0;\ h_y(2,3) \leftarrow 0$

$h_y(3,1) \leftarrow -1;\ h_y(3,2) \leftarrow -1;\ h_y(3,3) \leftarrow -1$

; Convolución en la dirección x e y:

$for\ x = 2\ hasta\ N-1$

$for\ y = 2\ hasta\ M-1$

$$g_x(x-1,y-1) \leftarrow h_x(1,1)\cdot f(x-1,y-1) + h_x(1,2)\cdot f(x-1,y) +$$
$$h_x(1,2)\cdot f(x-1,y+1) + h_x(2,1)\cdot f(x,y-1) +$$
$$h_x(2,2)\cdot f(x,y) + h_x(2,3)\cdot f(x,y+1) +$$
$$h_x(3,1)\cdot f(x+1,y-1) + h_x(3,2)\cdot f(x+1,y) +$$
$$h_x(3,3)\cdot f(x+1,y+1)$$

$$g_y(x-1,y-1) \leftarrow h_y(1,1)\cdot f(x-1,y-1) + h_y(1,2)\cdot f(x-1,y) +$$
$$h_y(1,2)\cdot f(x-1,y+1) + h_y(2,1)\cdot f(x,y-1) +$$
$$h_y(2,2)\cdot f(x,y) + h_y(2,3)\cdot f(x,y+1) +$$
$$h_y(3,1)\cdot f(x+1,y-1) + h_y(3,2)\cdot f(x+1,y) +$$

$$h_y(3,3) \cdot f(x+1, y+1)$$

end

end

; Combinación de gradientes $\sqrt{D_2(x)^2 + D_2(y)^2}$ (Recuérdese que el tamaño de la imagen de salida $g(x,y)$ ha descendido en dos respecto a $f(x,y)$ debido al problema de los límites cuando realizamos convoluciones comentado en el ejercicio anterior).

for $x = 1$ *hasta* $N - 2$

for $y = 1$ *hasta* $M - 2$

$g(x,y) \leftarrow \sqrt{g_x(x,y)^2 + g_y(x,y)^2}$

end

end

Representar imagen $g(x,y)$

fin

Se recomienda al alumno las siguientes actividades:

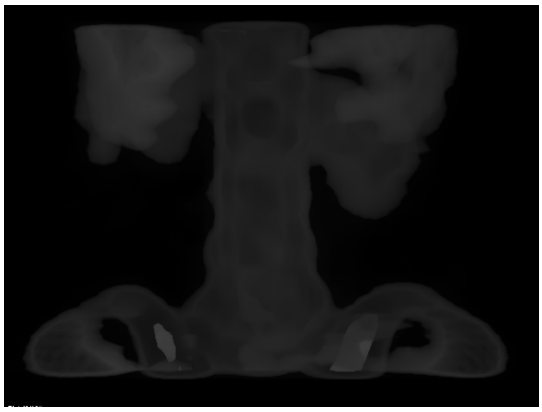
1. Implementar el algoritmo anterior relativo al operador de Prewitt en *scilab* probando su efecto sobre diferentes imágenes.
2. Estudiar el efecto de la aplicación de otras combinaciones de gradientes.
3. Realizar los pasos anteriores con otro tipo de operador (Sobel, Roberts, Laplaciana, Laplaciana de la gausiana).

PROBLEMA 4

Sea la imagen representada en la figura mediante valores del nivel de gris de 0 a 255. Codificar en pseudocódigo:

1. La transformación basada en el histograma más adecuada.

2. La técnica basada en la igualación del histograma.



¿Qué se pretende con este problema? Con este problema se intenta mostrar al alumno un ejemplo de programación de alguno de los métodos basados en la transformación del histograma.

Solución

1. *Transformación basada en el histograma*

   La imagen objeto de estudio se caracteriza por ser extremadamente oscura. Por consiguiente, una función cuadrática o una función cúbica será la más adecuada ya que con está se consigue dar un aspecto más blanquecino a la imagen.

   El pseudocódico del algoritmo que implementa la función cuadrática es:

   N: número de filas en la imagen.

   M: número de columnas en la imagen.

   ;Normalizar los niveles de gris:

   $for\ x = 1\ hasta\ N$

   $for\ y = 1\ hasta\ M$

   $f_N(x,y) \leftarrow f(x,y) * \frac{1}{255}$

   end

end

; Función raíz cuadrada:

$for\ x = 1\ hasta\ N$

$for\ y = 1\ hasta\ M$

$g_N(x,y) \leftarrow (f_N(x,y))^{\frac{1}{2}}$

end

end

; Restaurar los valores del nivel de gris de la imagen de salida:

$for\ x = 1\ hasta\ N$

$for\ y = 1\ hasta\ M$

$g(x,y) \leftarrow g_N(x,y) * 255$

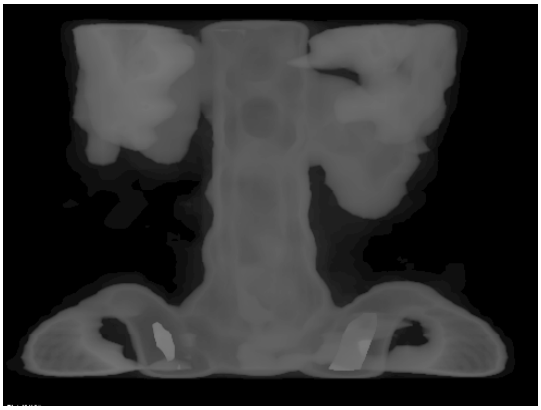
end

end

Representar imagen $g(x,y)$

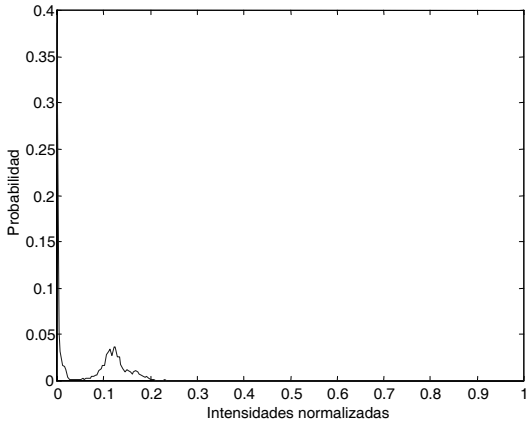
fin

La imagen resultante cuando se aplica la función raíz cuadrada se representa en la figura.



2. *Igualación del histograma*

   Si se analiza el histograma normalizado de la imagen se observa como $p_r(r)$ presenta valores distintos de cero para valores de r muy pequeños, esto es, la imagen es oscura.



Histograma normalizado

El propósito de esta transformación es conseguir una imagen con una distribución lo más homogénea posible. El pseudocódigo del método expuesto en la teoría es:

N: número de filas en la imagen.

M: número de columnas en la imagen.

n: número de niveles de gris diferentes (normalmente 255)

; Inicializar $p_r(r)$:

$for\ i = 1\ hasta\ n$

$\quad p_r(i) \leftarrow 0$

end

;Calcular la probabilidad $p_r(r)$:

$for\ x = 1\ hasta\ N$

$\quad for\ y = 1\ hasta\ M$

$for\ i = 1\ hasta\ n$

$si\ f(x,y) == i$

$p_r(i) \leftarrow p_r(i) + \frac{1}{255}$

end

end

end

end

;Calcular la función acumulativa: s_k

$for\ k = 1\ hasta\ n$

$si\ k == 1$

$s_1(k) \leftarrow p_r(k)$

$sino$

$s_1(k) \leftarrow s_1(k-1) + p_r(k)$

end

end

; Aproximación al nivel de gris más cercano:

$for\ k = 1\ hasta\ n$

$s(k) \leftarrow nivel\ de\ gris\ más\ cercano\ a\ s_1(k)$

end

; Imagen transformada o de salida:

$for\ x = 1\ hasta\ N$

$for\ y = 1\ hasta\ M$

$k \leftarrow 1$

$mientras\ f(x,y) <> k\_ésimo\ nivel\ de\ gris$

$k \leftarrow k + 1$; avanzar

end

$g(x, y) \leftarrow s(k)$

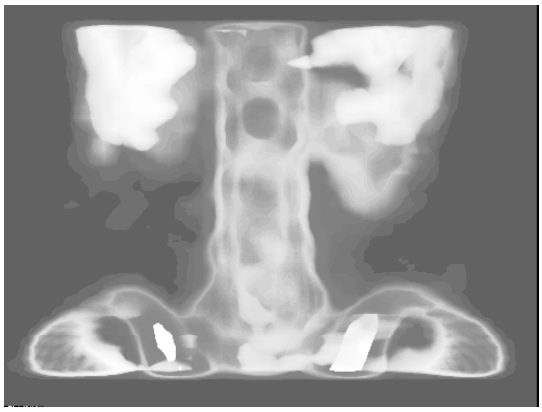
end

end

Representar imagen $g(x, y)$

fin

La imagen resultante se representa en la figura.



Se recomienda al alumno la siguiente actividad:

1. Implementar en *scilab* el algoritmo relativo a la igualación del histograma. Aplicarlo sobre diferentes imágenes.

Capítulo 3

Segmentación

Introducción y orientaciones para el estudio

El principal objetivo de la segmentación es dividir la imagen en regiones o zonas disjuntas que tengan correlación con objetos o zonas del mundo real. En este capítulo se van a describir los siguientes métodos:

- **Segmentación basada en la detección de fronteras**. Partiendo del conjunto de bordes detectados mediante cualquiera de los operadores estudiados en la detección de bordes el objetivo es asociarlos adecuadamente para definir las fronteras de los objetos. Los distintos métodos se diferencian según la cantidad de conocimiento que se tenga de la operación de agrupamiento de los bordes en contornos: análisis local en donde, en general se dispone de poco conocimiento y transformada de Hough en donde se conoce la ecuación paramétrica del contorno del objeto.

- **Segmentación basada en la umbralización**. La forma más simple de identificar regiones en una imagen es realizando una operación de umbralización, donde a un píxel se le asigna el valor de 1 si su nivel de gris supera un valor umbral y de 0 en caso contrario. El principal problema surge en la selección adecuada del valor del umbral. Una forma de conseguirlo es a través del estudio del histograma. En este punto se estudiarán las distintas técnicas de

umbralización basadas en el histograma.

- **Segmentación basada en el agrupamiento de píxeles**. Las técnicas de agrupación de píxeles son equivalentes a las técnicas de agrupación de datos caracterizadas por tratarse de técnicas no supervisadas, es decir, no existe un conocimiento previo de las clases de objetos existentes. La única información necesaria es la definición inicial de un vector de características. Algunos de estos algoritmos precisan conocer el número de clases existentes en la imagen. Este apartado está dedicado al estudio de dichas técnicas.

Tanto para el alumno de Ingeniería de Sistemas como de Gestión este capítulo es completamente nuevo. Para su estudio es necesario que este habituado con la terminología usada en visión y con el histograma de la imagen teniendo muy clara la información que éste representa además de conocer los métodos para la detección de bordes estudiados en el capítulo anterior. Se recomienda durante el estudio que todas las técnicas descritas a lo largo de este tema, una vez comprendidas, sean probadas sobre diferentes imágenes utilizando para ello el software recomendado.

Objetivos

Conocer los procesos y tareas empleados en la etapa de segmentación.

3.1 Introducción

La segmentación es una etapa fundamental en cualquier sistema de visión por computador, por las dificultades que conlleva así como por la importancia de sus resultados. Su principal objetivo es dividir la imagen en regiones o zonas disjuntas que tengan correlación con objetos o zonas del mundo real. Para ello se utilizarán características tales como el nivel de gris, color, textura, bordes o movimiento. En parte, esta etapa está muy unida a la etapa de reconocimiento ya que tras la segmentación, los objetos se encuentran perfectamente localizados en la imagen para a continuación ser reconocidos mediante su comparación con modelos.

La figura 3.1 refleja el objetivo del proceso de segmentación. Se observa cómo partiendo de una imagen definida por las intensidades del nivel de gris de cada uno de sus píxeles y previamente tratada en la etapa de preproceso, se ha obtenido una imagen dividida en diferentes regiones definidas a través de sus contornos o fronteras.

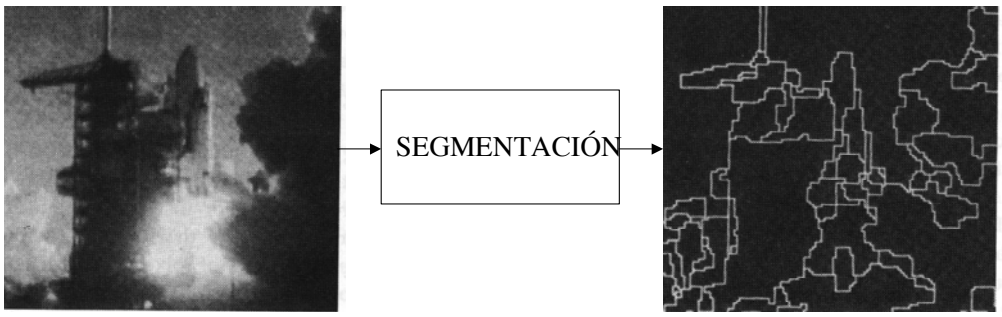


Figura 3.1: Etapa de segmentación.

A pesar de invertir un esfuerzo considerable en la detección de bordes y regiones, la segmentación sigue siendo muy difícil. En la detección de los bordes, por ejemplo, un umbral bajo permite detectar los bordes de bajo contraste pero también detecta los bordes falsos debido a las variaciones de las superficies; un umbral más alto es menos sensible tanto al ruido como a los verdaderos bordes. En el análisis de regiones, una región puede corresponder a más de una superficie, o por contra las variaciones en el nivel de gris de las superficies debidas a la iluminación pueden di-

vidir una superficie única en varias. Estos problemas pueden minimizarse buscando elementos significativos y ajustando varios parámetros de umbral. Pero la obtención de elementos significativos no puede resolverse sin un conocimiento externo posiblemente heurístico.

La cuestión está en hasta qué punto el proceso de detección de regiones debe ser dirigido por datos o por hipótesis. Es evidente que el conocimiento de los niveles más abstractos debe influir a los más primitivos. El uso de este conocimiento es muy complejo y necesita elaborados mecanismos para su control, dando lugar a implantaciones de sistemas muy especializados. Extender estos sistemas a dominios más amplios es difícil.

En general, el proceso de segmentación se basa en tres propiedades:

1. **Similitud**: Cada uno de los píxeles de un elemento tiene valores parecidos para alguna propiedad.

2. **Discontinuidad**: Los objetos destacan del entorno y, por tanto, tienen unos bordes definidos.

3. **Conectividad**: Los píxeles pertenecientes al mismo objeto tienen que ser contiguos, es decir, deben estar agrupados.

Estas suposiciones son difíciles de cumplir y sin embargo son imprescindibles para una buena segmentación. Respecto a la primera suposición, la similitud, los objetos deberían presentar una apariencia uniforme pero, en general, esto no es posible debido, entre otras causas a que la iluminación no sea constante existiendo pequeñas variaciones en el material, a la diferencia entre las ganancias para cada píxel de la cámara, a la existencia de brillos o a la presencia de ruido en la imagen. Respecto a la segunda condición los bordes no van a estar siempre bien definidos, por ejemplo en aquellos casos en los que el objeto y el fondo presenten apariencias semejantes. Por último se puede dar el caso de ocultamiento parcial de un objeto por parte de otro lo que provoca el incumplimiento de la tercera condición. De hecho, la

importancia de la iluminación en las aplicaciones industriales radica en que con ella se puede simplificar notablemente la etapa de segmentación ya que si ésta se utiliza adecuadamente es posible resaltar en algunos casos determinadas características de los objetos además de evitar problemas de brillos o efectos indeseados debido a la superficie de los objetos.

Teniendo en cuenta estas propiedades, las técnicas para obtener la segmentación se pueden basar en la búsqueda de las partes uniformes de la imagen o justo lo contrario, aquellas partes donde se produce un cambio. Dependerá del caso concreto el que se utilice un método u otro o la conjugación de ambos.

En este tema se van a describir diferentes métodos aplicados en la etapa de segmentación. Se ha de resaltar que no existe un método ideal para resolver los problemas. La aplicación de uno u otro método dependerá del problema que nos atañe así como del conocimiento del medio que se disponga. Atendiendo a las propiedades utilizadas para la realización del proceso de segmentación podemos clasificar los métodos en los siguientes grupos:

1. Segmentación basada en la detección de fronteras.
2. Segmentación basada en la umbralización.
3. Segmentación basada en el agrupamiento de píxeles.

Las dos últimas enfocan la segmentación como un problema de clasificación de píxeles, donde: píxeles de una misma región deben ser similares; píxeles de regiones distintas deben ser no similares; las regiones resultantes deben tener cierto significado para el procesamiento posterior. Hay que comentar que la similitud entre píxeles puede residir en el nivel de gris, textura, proximidad, posición en la imagen, etc. En cualquier caso, estas técnicas no son excluyentes, es decir, éstas pueden cooperar con el fin de conseguir una segmentación más perfecta.

3.2 Segmentación basada en la detección de fronteras

El cálculo de las fronteras puede plantearse, en principio, como una operación de filtrado de la imagen original puesto que los píxeles situados en los bordes de un objeto presentan grandes variaciones del nivel de gris en zonas muy pequeñas de la imagen. Dicho cálculo podría ser realizado durante la etapa de preproceso utilizando cualquiera de los detectores de bordes estudiados en el capítulo anterior (el operador de Prewitt o Sobel entre otros). Sin embargo, los bordes resultantes están desconectados, es decir, no se conoce la relación existente entre ellos además de que, como consecuencia del ruido, iluminación no uniforme y otros factores, estos operadores rara vez identifican por completo todos los bordes que constituyen la frontera de los objetos, necesitando, en tal caso, algoritmos que se encarguen de realizar la unión de los píxeles detectados.

La amplitud del problema se puede apreciar en la figura 3.2 donde se muestran los bordes detectados en la etapa de preproceso de una radiografía del pulmón.

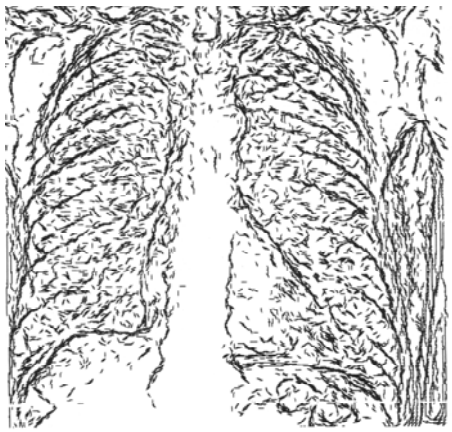


Figura 3.2: Radiografía del pulmón (Ballard y Brown 1982).

Los distintos métodos se diferencian según la cantidad de conocimiento que se tenga de la operación de agrupamiento de los bordes en contornos. Se entiende

por conocimiento a las restricciones implícitas o explícitas sobre la probabilidad de un agrupamiento. Tales restricciones podrían proceder de consideraciones físicas generales o, más frecuentemente, de restricciones más fuertes puestas a la imagen como resultado de ciertas consideraciones del dominio. Si hay mucho conocimiento implica que la forma global del contorno y su relación con otras estructuras de la imagen está muy precisada. Sin embargo, si se dispone de poco conocimiento significa que la segmentación debe llevarse a cabo más sobre las pistas locales y suposiciones más generales.

Las restricciones pueden tomar muchas formas. El conocimiento de donde se espera encontrar un contorno permite crear restricciones para verificar su localización. En muchas ocasiones, el conocimiento del dominio determina el tipo de curva además del posible ruido. Así, por ejemplo, en imágenes de objetos poliédricos únicamente existirán contornos rectos. En cambio si se dispone de menos conocimiento del contenido de la imagen quizás sea necesario recurrir al conocimiento del mundo a partir del cual es posible obtener restricciones que son verdad para la mayoría de los dominios. Por ejemplo, en ausencia de evidencias que muestren lo contrario, la conexión entre dos puntos se considera que es la línea más corta que los une. Esta clase de principios generales se utiliza en los algoritmos de segmentación para encontrar los contornos de los objetos que aparecen en la imagen.

Los métodos aplicados a la detección de fronteras que van a ser objeto de estudio en esta sección son:

1. Análisis local.

2. Análisis global mediante la transformada de Hough.

Estos métodos presentan los inconvenientes comunes siguientes: (1) al intentar eliminar los efectos del ruido durante la etapa de preproceso también se eliminan parte de los bordes de los objetos. Así, éstos no podrán ser detectados durante el proceso de detección de bordes apareciendo discontinuidades que dificultan la defi-

nición de los contornos de los objetos pudiendo llevar, en algunos casos, a resultados erróneos ; (2) suelen aproximar los puntos detectados a curvas de forma conocida y en consecuencia el resultado de la segmentación es siempre una aproximación del contorno de los objetos.

3.2.1 Análisis local

Una de las aproximaciones más simples para detectar si un píxel pertenece o no a un contorno consiste en analizar las características de los píxeles existentes en su vecindad (3×3 ó 5×5). Todos los píxeles con propiedades comunes serán asociados para formar un límite. Por ejemplo, podría obtenerse el contorno o frontera de un objeto a partir de píxeles con propiedades similares en la imagen del gradiente. En este caso, las dos propiedades utilizadas para establecer la similitud entre los píxeles del contorno son:

1. *La magnitud del gradiente:*

   Se dice que un píxel (x', y') es similar en magnitud al píxel vecino (x, y), si se cumple:

   $$|G[f(x, y)] - G[f(x', y')]| \leq U \tag{3.1}$$

   siendo U el valor del umbral para la diferencia de magnitud y G una combinación de los operadores gradiente relativos a la dirección X e Y. Para el cálculo del gradiente es posible aplicar cualquiera de los métodos expuestos en el capítulo anterior.

2. *La dirección del gradiente:*

   Se dice que el píxel (x', y') tiene un ángulo similar al píxel vecino (x, y), si:

   $$|\alpha(x, y) - \alpha(x', y') < A| \tag{3.2}$$

   donde A es el valor del umbral para la diferencia de ángulo.

Recuérdese que la dirección del vector gradiente se puede calcular aplicando la expresión:

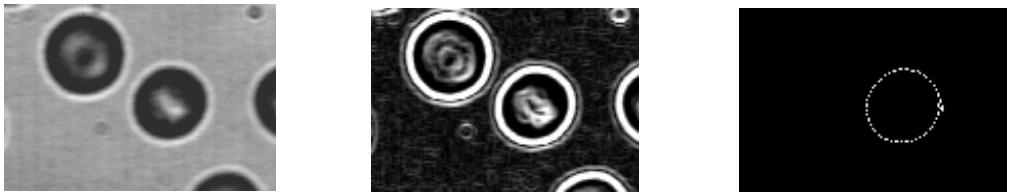
$$\alpha(x,y) = tag^{-1}\frac{G_y[f(x,y)]}{G_x[f(x,y)]} \tag{3.3}$$

siendo G_x y G_y el operador gradiente en las direcciones X e Y, respectivamente.

Nótese que la dirección del borde (x,y) es en realidad perpendicular a la dirección del vector gradiente en ese punto. Sin embargo, para el propósito de comparar direcciones, la ecuación (3.3) produce resultados equivalentes.

Partiendo de un píxel que pertenece al contorno del objeto a segmentar, la idea de esta técnica consiste, básicamente, en ir extendiendo la frontera en la dirección perpendicular al gradiente con píxeles vecinos que cumplan tanto el criterio de magnitud como el criterio de dirección del gradiente. Este tipo de técnica recibe el nombre de *seguimiento del contorno*. En la figura 3.3 se muestra un ejemplo de aplicación de esta técnica.

La selección del primer punto es crítica y se hace tanto teniendo en cuenta el valor del vector gradiente como cualquier información a priori que se tenga sobre el objeto a segmentar. Además, dado que los algoritmos de seguimiento de contorno no van a detectar necesariamente una curva cerrada, se precisa seguir el contorno en los dos sentidos del borde, esto es, $+90$ y -90 grados del vector gradiente. Así, el algoritmo avanzará en alguna de estas direcciones incorporando aquellos píxeles que satisfagan la condición de similitud.



(a) Imagen original

(b) Detección de bordes con el operador Sobel

(c) Segmentación de uno de uno de los objetos

Figura 3.3: Aplicación del algoritmo de análisis local

Tal y como se ha comentado anteriormente debido a determinados factores (ruido, iluminación, etc) pueden existir saltos entre los bordes detectados por el detector de bordes. Por este motivo, el proceso de incorporación de píxeles a la frontera suele emplear dos procedimientos complementarios: el seguimiento del contorno y el salto y búsqueda de nuevos píxeles de comienzo. El primer procedimiento considera como candidatos los vecinos del último píxel incorporado en la dirección del contorno. De entre éstos, se selecciona como nuevo candidato aquél cuyo gradiente sea máximo, siempre y cuando cumpla la condición de similitud. En el caso de que no se encuentren píxeles vecinos que no verifiquen la condición de similitud, se procede a la ejecución del segundo procedimiento caracterizado por buscar nuevos píxeles a lo largo de la frontera que verifiquen esta condición. Así, por ejemplo, la primera estrategia puede consistir en dejar un hueco de un píxel en la dirección del contorno. Si en esta zona tampoco se encuentran píxeles que verifiquen la condición de similitud se puede dejar un hueco de dos o más píxeles. Si este procedimiento de búsqueda encuentra un píxel de la frontera, los píxeles del hueco en la dirección del borde son también incorporados y se continúa el seguimiento del contorno a partir de este último píxel. Si por el contrario, no se encuentra ningún píxel que pueda ser incorporado, el procedimiento de búsqueda en esa dirección ha terminado. Seguidamente se procede de manera análoga en la otra dirección. Por tanto, el proceso finaliza bien cuando los procedimientos anteriores no consiguen incorporar píxeles a la frontera, o bien cuando el punto original de comienzo del algoritmo es reencontrado.

Es importante comentar que debido al ancho en la respuesta del operador detector de bordes, puede ocurrir que en fronteras cerradas se detecten contornos paralelos muy próximos. Por este motivo, los píxeles situados en la dirección normal al contorno se eliminan conforme el algoritmo incorpora un nuevo píxel. Esto se puede realizar, por ejemplo, etiquetándolos de tal manera que queden excluidos como candidatos en posteriores iteraciones.

Para el análisis de los bordes puede resultar de ayuda la teoría de grafos. Un grafo es un conjunto de nodos y arcos entre ellos. Por ello se pueden tomar los

puntos de borde como nodos de un grafo definiéndose una función de coste entre los puntos inicial y final x_1 y x_N. Por ejemplo la función:

$$C = -\sum_{i=1}^{N} |e(x_i)| + a\sum_{i=2}^{N} |\theta(x_i) - \theta(x_{i-1})| + b\sum_{i=2}^{N} |e(x_i) - e(x_{i-1})| \tag{3.4}$$

representa el coste desde el punto inicial a otro intermedio y desde éste hasta el final. Así, cuanto mayor sea el módulo del gradiente menor será el coste, mientras que la diferencia entre los módulos y dirección del gradiente penalizan el coste. Este coste debe ser calculado para cada uno de los posibles caminos eligiéndose aquél cuyo coste sea menor. Aunque los resultados obtenidos con esta función son buenos es complicado de implementar ya que han de tenerse en cuanta todos los posibles caminos además puede darse el caso de que cuanto más corto sea el camino menor sea su coste lo que no implica que sea el contorno del objeto. Por último se necesita saber el punto inicial y final lo cual no siempre es posible dado que el algoritmo parte simplemente de un conjunto de puntos detectados como posibles bordes y pertenecientes a todos los contornos de los objetos existentes en la imagen.

Como alternativa se tiene la aplicación de la programación dinámica. Ésta busca el valor óptimo de una función en la que no todas las variables están relacionadas simultáneamente. En el presente caso se supone que el camino óptimo entre dos puntos tiene subcaminos óptimos para los puntos intermedios. De esta forma se podría definir una función de coste similar a la anterior pero que busque el camino que obtenga el valor más alto:

$$C = \sum_{i=1}^{N} |e(x_i)| - a\sum_{i=2}^{N} |\theta(x_i) - \theta(x_{i-1})| \tag{3.5}$$

función que puede expresarse de forma recursiva como:

$$C(x_1, x_N) = C(x_1, x_{N-1}) + |e(x_N)| - a\,|\theta(x_N) - \theta(x_{N-1})| \tag{3.6}$$

Entonces el camino óptimo puede dividirse en dos subcaminos óptimos que satisfagan:

$$C_{\acute{o}ptimo}(x_1, x_N) = \max C(x_1, x_N) = \max C(x_1, x_{N-1}) + |e(x_N)| - a\,|\theta(x_N) - \theta(x_{N-1})| \tag{3.7}$$

$$C_{\acute{o}ptimo}(x_1, x_N) = C_{\acute{o}ptimo}(x_1, x_{N-1}) + \max\{|e(x_N)| - a\,|\theta(x_N) - \theta(x_{N-1})|\} \quad (3.8)$$

Luego a partir del primer píxel se buscará cada píxel que maximice el segundo sumando. Veamos el ejemplo extraído de [4] y mostrado en la figura 3.4 en donde se tienen tres píxeles que pueden ser el punto inicial del camino (*a, b, c*), otros tres que pueden ser el punto final (*g, h, i*) y los píxeles intermedios (*d, e, f*); además se representa el valor de pasar de uno a otro. Si se quiere pasar por d el camino óptimo sería *a*-*d* al tener el valor mayor, para el punto e es *b*-*e* y para el punto f es *c*-*f*. Con esto se tendrían los costes asociados a los puntos intermedios:

$$C(a, d) = 7$$

$$C(b, e) = 6$$

$$C(c, f) = 5$$

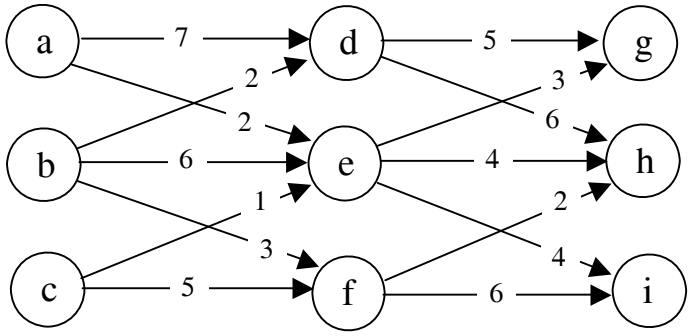


Figura 3.4: Determinación del camino mediante programación dinámica (Arturo de la Escalera, 2001).

Para pasar a los puntos finales se vería que para llegar a g el camino es *d*-*g*, para *h*, *d*-*h* y para i es *f*-*i*, con los costes:

$$C(d, g) = 5$$

$$C(d, h) = 6$$

$$C(f, i) = 6$$

Luego el coste total es:

$$C(a,g) = C(a,d) + C(d,g) = 7 + 5 = 12$$

$$C(a,h) = C(a,d) + C(d,h) = 7 + 6 = 13$$

$$C(c,i) = C(c,f) + C(f,i) = 5 + 6 = 11$$

Por lo que el camino a seguir pasa por los puntos *a*-*d*-*h*.

3.2.2 Análisis global mediante la transformada de Hough

Esta técnica es aplicable, en principio, siempre que se tenga un conocimiento aproximado de la localización del contorno, y su forma pueda ser descrita mediante una curva paramétrica. No obstante, puede ser empleada en la detección de curvas arbitrarias de las que no se dispone de su ecuación paramétrica pero se conoce su forma particular. Su principal ventaja es que no le afecta ni las lagunas existentes en la frontera del objeto ni el ruido.

A la hora de aplicar la transformada de Hough a una imagen es necesario obtener primero una imagen binaria de los píxeles que, previamente, forman parte de la frontera. Para ello, a partir de la imagen original se obtiene la imagen gradiente y luego se aplica un umbral de corte.

Para introducir el método comenzamos considerando el problema de detección de líneas rectas en la imagen. Supongamos que mediante la aplicación de algún proceso, ciertos puntos de la imagen han sido seleccionados por tener alta probabilidad de pertenecer a un contorno recto. La técnica de Hough organiza estos puntos en líneas rectas, básicamente considerando la totalidad de líneas rectas y seleccionando aquéllas que mejor se adaptan a los datos.

Sea el punto $(x',\ y')$ de la figura 3.5(a) y la ecuación de una recta definida mediante:

$$y' = mx' + c \tag{3.9}$$

¿Cuales son las líneas rectas que podrían pasar por (x', y')?. La respuesta es simple, todas las líneas con m y c que cumplan $y' = mx' + c$. Suponiendo (x', y') fijo, la última ecuación puede interpretarse como una línea en el espacio $m - c$ o espacio paramétrico. Repitiendo este razonamiento para un segundo punto (x'', y'') también obtendríamos una línea asociada en el espacio paramétrico y además estas líneas se cortarían en el punto (m', c') correspondiendo con la línea AB en el espacio imagen que conecta ambos puntos. En efecto, todos los puntos sobre la línea AB producirán líneas en el espacio paramétrico que se cruzan en el punto (m', c'), como muestra la figura 3.5(b).

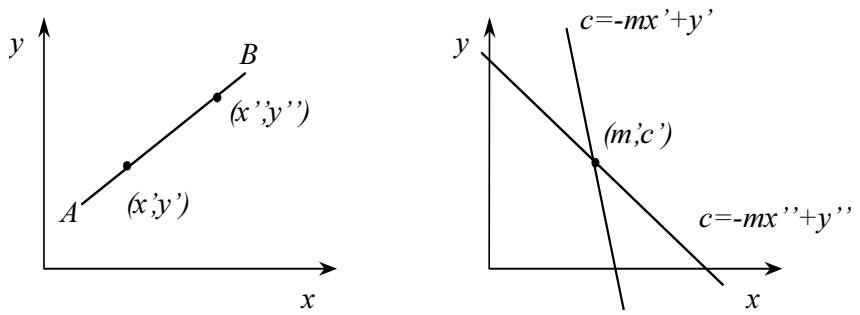


Figura 3.5: Transformación de Hough.

Para aplicar la transformada de Hough es necesario discretizar el espacio de parámetros en una serie de celdas denominadas de acumulación. Esta discretización se realiza sobre los intervalos $(m_{\min}, m_{\max})$ y $(c_{\min}, c_{\max})$ que definen el conjunto de rectas buscadas, dando lugar a lo que se denomina acumulador.

Pues bien, de un modo mas general la idea clave del método es la siguiente. Supongamos que se dispone de un conjunto de puntos (o de un conjunto de características locales tales como un conjunto de puntos con sus tangentes asociadas) y además se conoce la forma paramétrica de la curva a la que pertenecen dichos puntos. Es posible determinar los parámetros de la curva siguiendo el siguiente procedimiento.

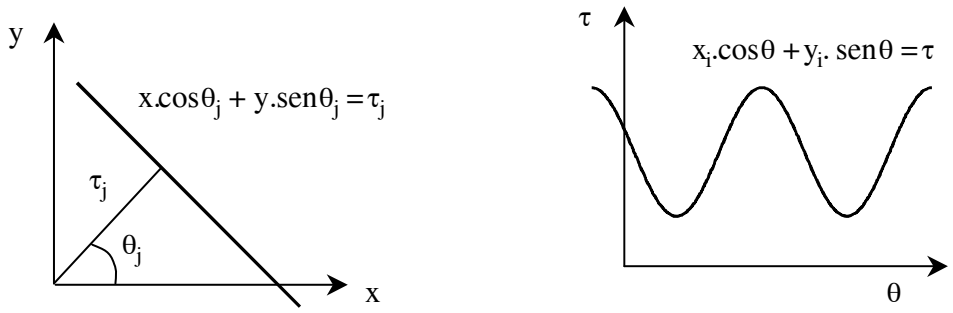
En primer lugar emparejamos cada punto, a través de la transformación de

Hough, con una de las celdas definidas en el espacio paramétrico. Lógicamente, los valores de c obtenidos tienen que aproximarse al valor discreto más cercano. Así, si al tomar m_i se obtiene c_j, se incrementa en uno el número de votos de la celda $A(i,j)$. Al final del proceso, el número de votos obtenidos en cada celda indica el número de puntos que, salvo errores de discretización, satisfacen la ecuación de la recta correspondiente. Por tanto, las celdas con mayor acumulación de votos constituyen el conjunto de rectas detectadas.

Un problema que aparece al usar la ecuación (3.9) para representar una recta es que, tanto la pendiente como la ordenada en el origen tienden a infinito conforme la recta se aproxima a posiciones verticales. Para evitar las singularidades que pueden provocar las rectas de pendiente infinita se usa la representación normal de una recta, dada por:

$$x \cdot cos\ \theta + y \cdot sen\ \theta = \tau \tag{3.10}$$

donde τ es la distancia de la recta al origen y θ es el ángulo que forma la normal con el eje X (ver figura 3.6).



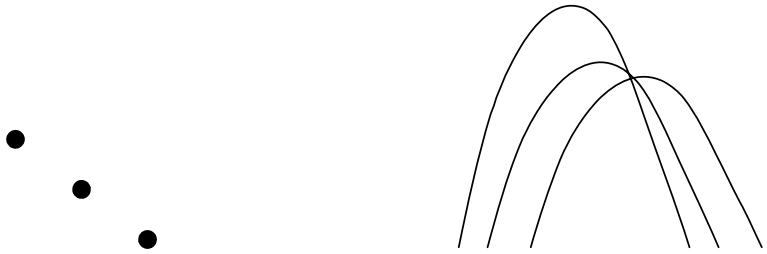
(a)Representación de una recta (b)Transformada de Hough del punto (x_i, y_i)

Figura 3.6: Transformada de Hough de una recta

La forma de construir el acumulador en el plano $\theta - \tau$ es similar a la expuesta, la única diferencia estriba en que, en lugar de líneas rectas, se obtendrán curvas sinusoidales. Así, M puntos colineales pertenecientes a una recta:

$$x \cdot cos\ \theta_j + y \cdot sen\ \theta_j = \tau_j \tag{3.11}$$

darán lugar a M curvas sinusoidales que se cortarán, en el espacio de parámetros, en el punto (θ_j, τ_j) como ilustra la figura 3.7.



(a) Imagen con 3 puntos alineados (b)Transformada de Hough correspondiente

Figura 3.7: Transformada de Hough de una recta

Sin embargo, este método presenta como inconveniente la cantidad de cálculo necesario cuando el número de parámetros que definen la curva aumenta, haciendo esta técnica solamente práctica para curvas con un número pequeño de parámetros. Una solución para simplificar el problema consiste en el uso de características asociadas a cada punto. Veamos un ejemplo.

Consideremos, en principio, un conjunto de puntos pertenecientes a un círculo cuyos parámetros se desea determinar. La ecuación paramétrica de un círculo viene dada por:

$$(x-a)^2 + (y-b)^2 = r^2 \tag{3.12}$$

donde se observa la presencia de tres parámetros a, b y r. Por lo tanto, el acumulador será tridimensional, es decir, presenta la forma $A(i, j, k)$.

El procedimiento consiste en incrementar a y b, determinar r mediante la ecuación (3.12) y votar a la celda asociada. Lógicamente esta tercera dimensión exige un número de operaciones mucho mayor.

Supongamos ahora un conjunto de puntos, cada punto con una tangente asociada, pertenecientes a un círculo cuyos parámetros se desea determinar. Un círculo tiene tres grados de libertad: dos para la posición de su centro (a, b) y uno para su radio r. Sin embargo, cualquier círculo que pase a través de un punto fijo con su

tangente conocida tiene únicamente un grado de libertad: como muestra la figura 3.8, el centro del círculo debe yacer sobre la línea ortogonal a la tangente y pasar a través del punto. Se observa cómo fijando el centro del círculo, que estará en dicha línea ortogonal, determinamos su radio. De aquí que dado un punto con una tangente asociada, los tres parámetros del círculo se reducen a un grado de libertad en el espacio paramétrico.

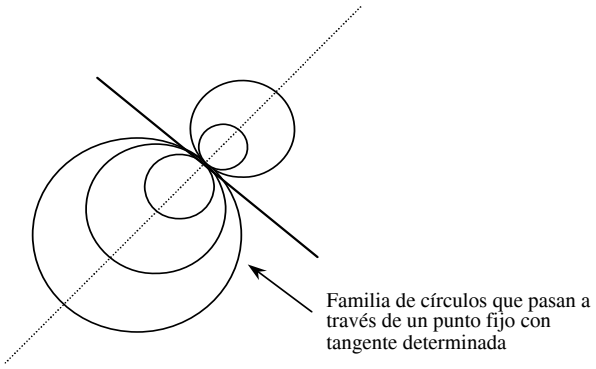


Figura 3.8: Reducción del grado de libertad en la transformación de Hough para el caso de un círculo.

Sea el punto (x_0, y_0) y el ángulo de su tangente asociada θ_0, entonces la transformación de Hough viene dada por:

$$\begin{aligned} x_0 &= x_c + r\text{sen}\theta_0 \\ y_0 &= y_c + r\cos\theta_0 \end{aligned} \tag{3.13}$$

donde (x_c, y_c) son las coordenadas del centro del círculo y r es el radio del círculo. De forma similar, otro punto generará otra curva. Los parámetros correspondientes al círculo que buscamos se determinan mediante la intersección de todas estas curvas.

Puede darse la circunstancia de que el conjunto de curvas no tenga una intersección común, entonces no existe una curva definida en forma paramétrica que satisfaga los datos dados. Análogamente si la intersección no es un punto, entonces la solución es ambigua. Finalmente, se ha de destacar que los parámetros calculados podrían definir una curva infinita (una línea, una parábola). Por tanto, si

únicamente se desea un segmento de curva será necesario un método adicional para especificar los puntos extremos.

La gran ventaja de la transformada de Hough es que utiliza toda la información de la imagen por lo que es más inmune a pérdidas parciales de los bordes que los métodos locales. Sus inconvenientes son el coste computacional mencionado anteriormente y para el caso de las rectas es que las rectas detectadas son de longitud infinita. Es decir no se sabe donde empieza y donde acaba el segmento de recta presente en la imagen lo que puede provocar la elección de varios segmentos como pertenecientes al mismo objeto cuando en realidad pertenecen a objetos distintos. Por ejemplo en la figura 3.9 es difícil determinar el número y posición de los triángulos presentes en la imagen a partir de la información obtenida mediante la transformada de Hough.

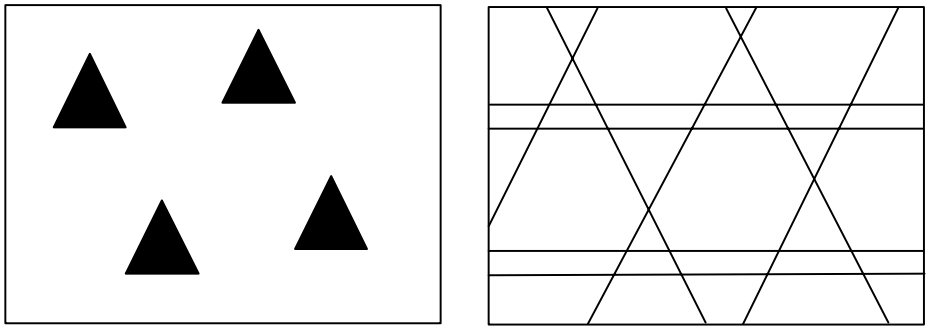


Figura 3.9: Problemas de la transformada de Hough para rectas.

Caso general

Consideremos ahora el caso general en el que los objetos no tienen una forma analítica simple, pero poseen una silueta particular.

La idea del método consiste en utilizar una tabla de consulta para definir la relación existente entre los píxeles de la frontera y los parámetros del espacio de Hough. Lógicamente, los valores de la tabla han de ser obtenidos previamente utilizando una forma prototipo.

Supongamos que el objeto que aparece en la figura 3.10 tiene forma conocida. El método comienza construyendo la tabla para lo cual, en primer lugar, se selecciona un punto de referencia (x_{ref}, y_{ref}) a partir del cual se definirá la forma del objeto.

El siguiente paso es calcular para cada punto de la frontera (x_i, y_i), la orientación α_i, la distancia r_i y la dirección β_i con respecto al punto de referencia (ver figura 3.10). A continuación los valores (r_i, β_i), indexados con su α_i correspondiente se almacenan en una tabla denominada *R-tabla*. Dado que es posible encontrar más de una vez una determinada orientación α_i, hay que prever en la tabla más de un registro de distancia y dirección por cada valor de la orientación. En la tabla 1.1 se muestra el aspecto de la *R-tabla.*

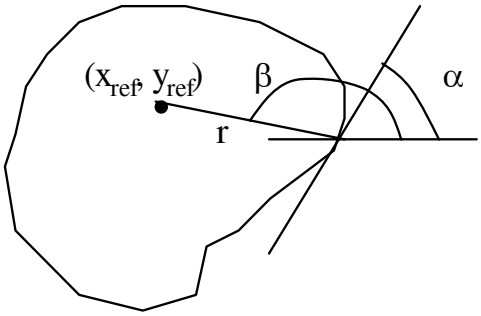


Figura 3.10: Geometría usada para obtener la R-tabla.

Ángulo medido	Conjunto de radios $r = (r, \beta)$
α_1	r_1^1, r_2^1, ..., $r_{n_1}^1$
α_2	r_1^2, r_2^2, ..., $r_{n_2}^2$
:	:
:	:
α_m	r_1^m, r_2^m, ..., $r_{n_m}^m$

Tabla 1.1 R tabla

Una vez construida la *R-tabla*, el espacio de Hough se define en función de la posibles posiciones de la forma en la imagen, función a su vez de las posibles posiciones del punto de referencia (x_{ref}, y_{ref}). Así, para cada píxel (x, y) de la imagen

se calculan las coordenadas de la celda a incrementar utilizando la expresión:

$$\begin{aligned} x_{ref} &= x + r \cdot cos\ \beta \\ y_{ref} &= y + r \cdot sen\ \beta \end{aligned} \tag{3.14}$$

Los valores de r y β a utilizar en la ecuación (3.14) se obtienen de la *R-tabla* a partir del valor de α calculado en el píxel (x, y) de la imagen. En el caso de que existan en la tabla varios pares de (r, β) para ese valor de α, se utilizan todos, incrementándose, diferentes celdas (x_{ref}, y_{ref}).

En todo lo comentado hasta el momento se ha considerado que es conocida la orientación del objeto. Si no fuera así, es necesario aumentar la dimensión del acumulador incorporando un parámetro adicional Ω que considere las posibles orientaciones del objeto. En este caso se empleará un acumulador tridimensional $(x_{ref}, y_{ref}, \Omega)$, que será incrementado utilizando las expresiones:

$$\begin{aligned} x_{ref} &= x + r \cdot cos\ (\beta + \Omega) \\ y_{ref} &= y + r \cdot sen\ (\beta + \Omega) \end{aligned} \tag{3.15}$$

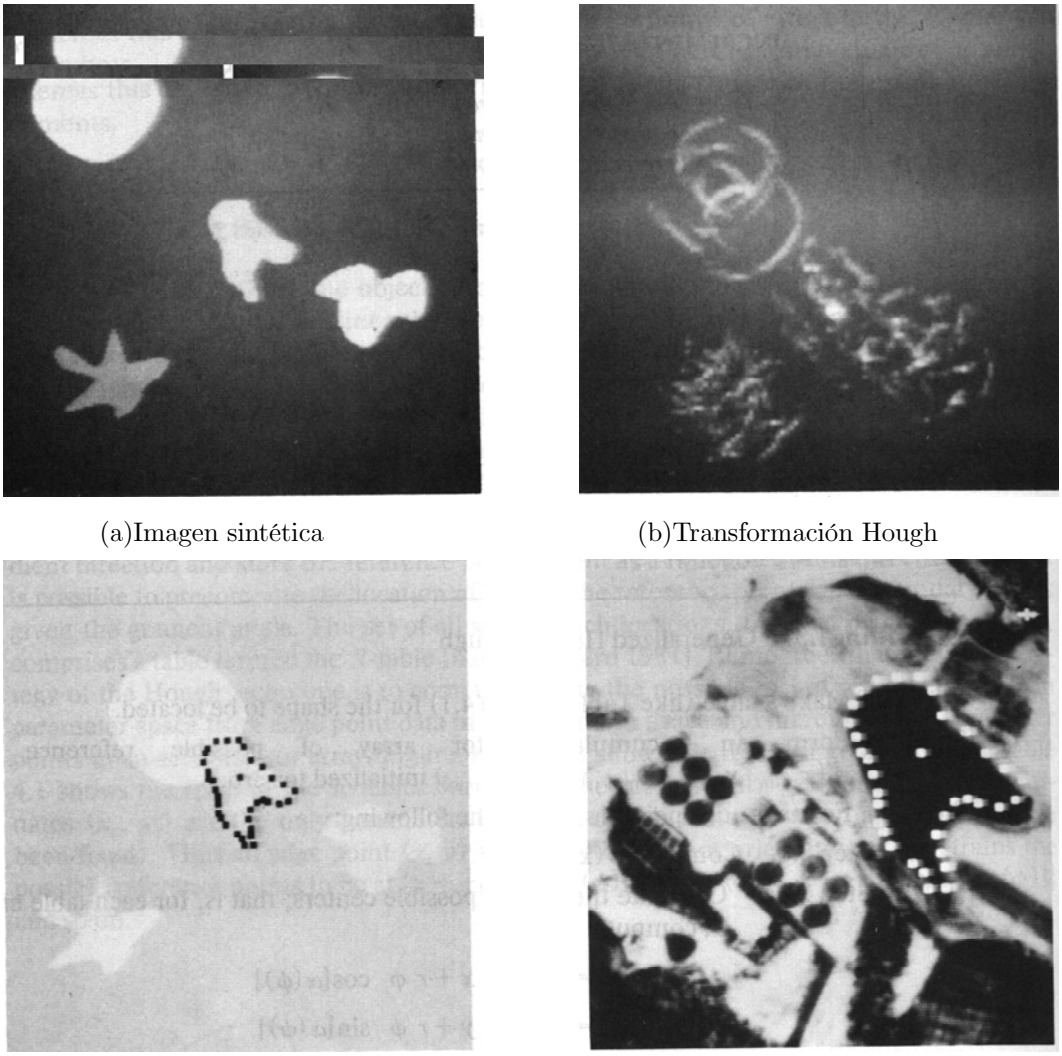
para todos los valores discretos considerados de Ω. Nótese que los valores de r y β serán obtenidos de la *R-tabla* entrando con el ángulo $\alpha - \Omega$. Los resultados obtenidos cuando se emplea esta transformación para detectar en la imagen un objeto con una silueta general se muestran en la figura 3.11.

3.3 Segmentación basada en la umbralización

Se van a estudiar dos métodos de umbralización aplicables en la etapa de segmentación:

1. Umbralización binaria.

2. Umbralización basada en el histograma



(a)Imagen sintética (b)Transformación Hough

(c)Forma detectada (d)La misma forma en otra imagen

Figura 3.11: Ejemplo de la aplicación de la técnica de Hough

3.3.1 Umbralización binaria

La forma más simple de identificar regiones en una imagen es realizando una operación de umbralización, donde a cada píxel se asigna el valor binario 1 si su nivel de gris es suficientemente alto, superando un valor umbral. Después de esta operación se obtienen imágenes binarias formadas por regiones compuestas por grupos de píxeles.

La determinación automática del valor del umbral es sencilla en los casos en que se conocen las distribuciones de píxeles claros y oscuros. El valor de separación ideal sería aquél para el cual las fracciones de píxeles claros y oscuros fueran iguales (la probabilidad de la asignación de un píxel a una zona sería igual para los dos grupos). Pero lo que se conoce en general no es esta distribución sino el histograma de la imagen. Precisamente, la umbralización basada en el histograma es el siguiente método de estudio.

3.3.2 Umbralización basada en el histograma

El concepto de histograma de una imagen digital ya ha sido estudiado en el tema anterior. El histograma nos permitía conocer la frecuencia relativa de aparición de cada uno de los posibles niveles de intensidad dentro de la imagen. Si consideramos el proceso de segmentar una imagen digital como la agrupación de los píxeles, el histograma servirá para agrupar los píxeles en función del valor de su intensidad.

Para entender la utilización del histograma durante en la umbralización comenzaremos por realizar el siguiente análisis. Supongamos la figura 3.12 en donde aparece una imagen sencilla constituida por un objeto (en este caso oscuro) y un fondo uniforme (en este ejemplo claro). En la misma figura se ha representado el histograma de la imagen. Este histograma es claramente bimodal es decir, presenta dos máximos y un único mínimo, ya que los píxeles de la imagen se distribuyen en dos clases o regiones: el objeto y el fondo. En este caso es muy sencillo determinar un valor umbral que discrimine de manera óptima las dos regiones existentes.

Otro ejemplo se muestra en la figura 3.13 en donde existen dos clases de objetos uno oscuro con forma cuadrada y otro intermedio con forma redonda ambos sobre un fondo claro. En el histograma se observa como éste no aporta información espacial es decir información sobre la posición que ocupa un determinado objeto y por tanto, los dos objetos idénticos en cuanto a sus valores de intensidad aparecen confundidos en el histograma.

Cuando la umbralización se basa exclusivamente en los valores de intensidad de los píxeles de la imagen se denomina umbralización global, en contraposición existe la umbralización local en la que aparte de esta propiedad global de un píxel se utiliza información local del mismo, es decir, propiedades de un píxel que dependen de su localización en la imagen. Pues bien, cuando el valor del umbral depende exclusivamente de $f(x, y)$ el umbral se denomina global. Si depende de $f(x, y)$ y de alguna propiedad local de ese punto se llama local. Si además depende de las coordenadas x e y se llama umbral dinámico.

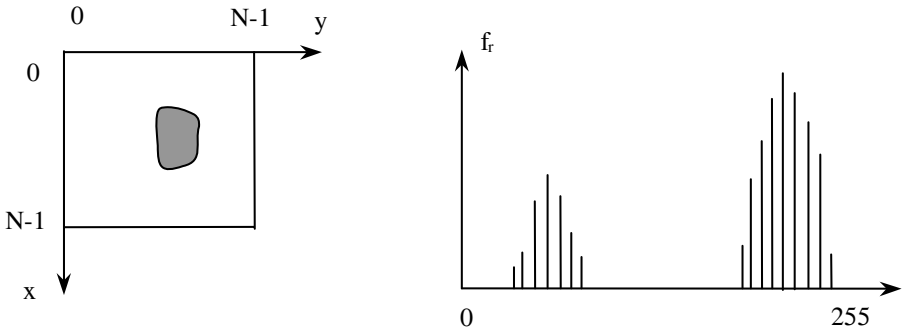


Figura 3.12: Imagen digital de un objeto oscuro sobre un fondo claro y su correspondiente histograma.

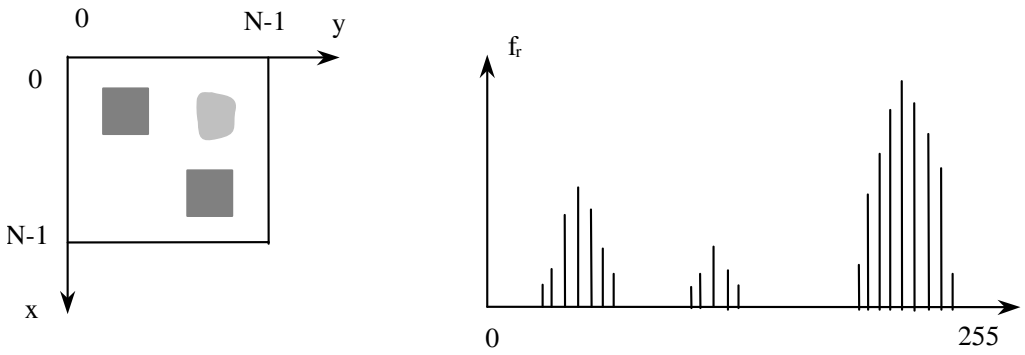


Figura 3.13: Imagen digital con dos objetos oscuros y un objeto de luminosidad intermedia sobre un fondo claro. El histograma presenta tres tramos cada uno de ello producido por un patrón o clase de objeto.

Técnicas de umbralización global

Los umbrales globales se utilizan cuando existe una clara definición entre los objetos y el fondo, y cuando la iluminación es relativamente uniforme. Por ejemplo, las técnicas que usan luz estructurada dan como resultado imágenes que pueden ser segmentadas mediante umbrales globales.

Respecto a la umbralización global existen varios procedimientos a la hora de implementar el algoritmo. Un análisis de distintas técnicas para la selección del umbral se presenta en [4]. En esta asignatura estudiaremos únicamente dos procedimientos extraídos de [7]:

1. Umbralización basada en la búsqueda de mínimos.
2. Umbralización basada en las técnicas de reconocimiento de patrones o formas.

Umbralización basada en la búsqueda de mínimos Se trata de un procedimiento heurístico cuyo objetivo es localizar niveles de intensidad que umbralicen adecuadamente el histograma de la imagen en estudio. Se ha visto en las figuras anteriores que los píxeles se agrupan aproximadamente, en trozos individuales dentro del histograma. Cada trozo está formado por los niveles de intensidad de los píxeles correspondientes a objetos físicos equivalentes es decir pertenecientes al mismo patrón. Dado que dos trozos consecutivos definen un valle de separación entre ellos el nivel de intensidad más bajo del valle, en principio, podría aplicarse como umbral de separación entre los dos trozos adyacentes.

El principal inconveniente de este método radica en la naturaleza ruidosa del histograma, en donde aparecen multitud de falsos mínimos y falsos máximos lo cual influye negativamente en la efectividad del algoritmo. Un ejemplo de esta idea se muestra en el histograma de la figura 3.14 en donde existen dos clases de objetos. Los dos máximos relativos que aparecen entre los dos trozos extremos no se tratan de otros dos tipos de objetos. Éstos pueden ser debidos a las condiciones de iluminación

y a los ruidos inherentes a la captación y digitalización de la imagen.

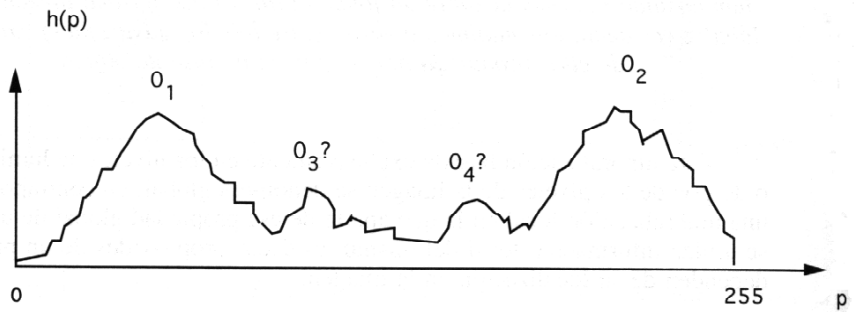


Figura 3.14: Ejemplo de un histograma con posible ruido.

Para detectar, y por tanto rechazar, estos falsos objetos, es necesario realizar un estudio previo, durante la fase de diseño del sistema final, de la distribución de los valores de intensidad de los posibles tipos de objetos a reconocer. Precisamente, era la etapa de preproceso la encargada de disminuir el ruido existente en el histograma. Así, la aplicación de un filtro de paso bajo convenientemente sintonizado a una frecuencia de corte adecuada, elimina las componentes de alta frecuencia del histograma; es decir, aquéllas producidas por cambios bruscos del histograma en intervalos reducidos.

Umbralización basada en las técnicas de reconocimiento de formas Si se analiza la forma que presentan los histogramas de las figuras 3.12 y 3.13, podrá observarse que los trozos que en ellas aparecen recuerdan a la campana de Gauss, típica de las distribuciones normales.

Pues bien, este segundo enfoque se basa en considerar que cada trozo del histograma corresponde a la distribución en probabilidad de un patrón. En sentido estricto esta suposición es válida, ya que el histograma $h(p)$ verifica las propiedades siguientes:

$$0 \leq h(p) \leq 1; \qquad \sum_{p=0}^{2^q-1} h(p) = 1 \tag{3.16}$$

Esto es, $h(p)$ cumple las dos restricciones de una función de densidad de probabilidad.

Bajo esta óptica es posible aplicar las técnicas bayesianas en el proceso de umbralización. Para centrar ideas vamos a trabajar en el caso más simple en el que existen solamente dos clases, tal como se muestra en el histograma de la figura 3.15 en donde se han aproximado los dos trozos del histograma real a dos campanas de Gauss; lo cual significa que se han modelado las dos clases α_1 y α_2 como variables aleatorias normales y gausianas..

Podemos escribir:

$$p(r/\alpha_1) = \frac{1}{\sqrt{2\pi}\sigma_1} \cdot e^{-\frac{(r-m_1)^2}{2\sigma_1^2}} \tag{3.17}$$

$$p(r/\alpha_2) = \frac{1}{\sqrt{2\pi}\sigma_2} \cdot e^{-\frac{(r-m_2)^2}{2\sigma_2^2}} \tag{3.18}$$

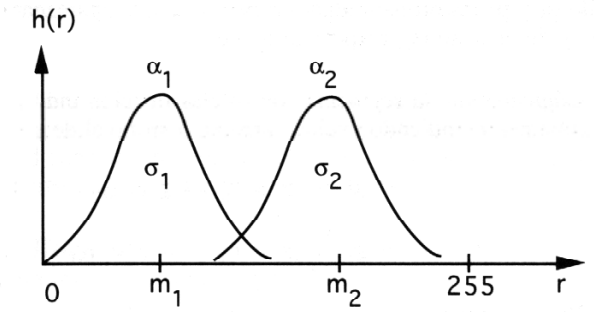


Figura 3.15: Histograma ideal de dos objetos en donde se han aproximado los trozos a dos campanas de Gauss.

Sabemos que el umbral óptimo r_u es aquél que se obtiene de la ecuación transcendente:

$$p_1\ p(r_u/\alpha_1) = p_2\ p(r_u/\alpha_2) \tag{3.19}$$

siendo p_1 y p_2 las probabilidades a priori de que aparezca un objeto de la clase α_1 y de la clase α_2, respectivamente.

Lógicamente, la segmentación de los píxeles de la imagen atendiendo únicamente

a sus intensidades será:

$$si\ z > r_u \rightarrow z \in \alpha_2$$

$$si\ z < r_u \rightarrow z \in \alpha_1$$

Para encontrar el valor de r_u que verifique la igualdad es necesario emplear algoritmos neperianos. En tal caso, el umbral óptimo cumple la ecuación cuadrática:

$$r_u^2 + a\ r_u + b = 0 \tag{3.20}$$

siendo los coeficientes a y b:

$$a = 2\ \frac{(m_1\ \sigma_2^2 - m_2\sigma_1^2)}{(\sigma_1^2 - \sigma_2^2)} \tag{3.21}$$

$$b = 2\ \frac{\sigma_1^2\ \sigma_2^2}{(\sigma_1^2 - \sigma_2^2)} \ln\ \frac{\sigma_2\ p_1}{\sigma_1\ p_2} - \frac{a}{2} \tag{3.22}$$

Si ambas clases presentan la misma dispersión estadística, $\sigma_1 = \sigma_2 = \sigma$, el umbral óptimo es entonces:

$$r_u = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln\ \frac{p_2}{p_1} \tag{3.23}$$

Si además, las clases son equiprobables:

$$r_u = \frac{(m_1 + m_2)}{2} \tag{3.24}$$

que concuerda con el concepto intuitivo de que el nivel óptimo de umbralización cuando las clases presentan la misma distribución de probabilidad (los trozos son exactamente iguales) es el punto medio entre las medias de las clases.

El problema práctico a la hora de segmentar una imagen con este método es calcular las medias y las desviaciones típicas de cada trozo. La principal dificultad consiste en delimitar el rango de los trozos; es decir, la amplitud de las correspondientes campanas de Gauss.

Generalizando el caso de n objetos, aparecerán tantas campanas de Gauss como clases de objetos (ver figura 3.16).

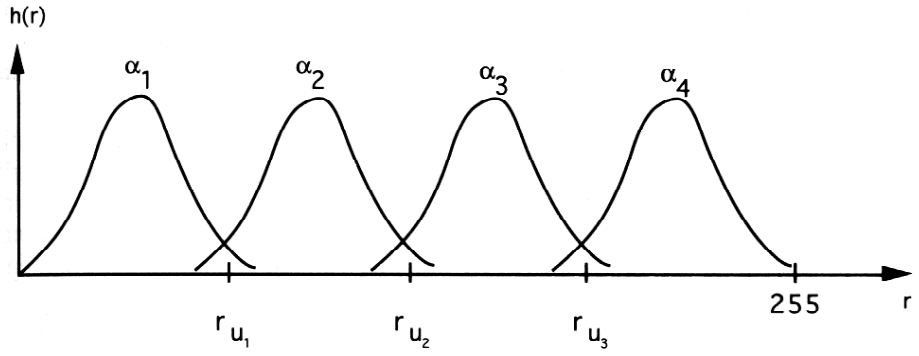


Figura 3.16: Histograma idealizado de una imagen con cuatro clases de objetos.

El umbral r_{u_1} discrimina los píxeles de la clase de objeto α_1 con el resto, r_{u_2} discrimina los píxeles de α_2 y, por último, r_{u_3} determina si pertenecen a α_3 o a α_4.

$$
\begin{aligned}
si\ Z &< r_{u_1} \rightarrow Z \in \alpha_1 \\
si\ (Z &> r_{u_1}) \cap (Z < r_{u_2}) \rightarrow Z \in \alpha_2 \\
si\ (Z &> r_{u_1}) \cap (Z > r_{u_2}) \cap (Z < r_{u_3}) \rightarrow Z \in \alpha_3 \\
si\ (Z &> r_{u_1}) \cap (Z > r_{u_2}) \cap (Z > r_{u_3}) \rightarrow Z \in \alpha_4
\end{aligned}
$$

Hasta ahora hemos planteado la umbralización del histograma bajo un enfoque global; es decir, manipulando la imagen digital completa. Este enfoque puede conducir a una segmentación incorrecta, especialmente si las condiciones de iluminación son defectuosas o cambiantes, de tal manera que los niveles de intensidad luminosa no se distribuyen de forma homogénea en la imagen.

Para evitar los problemas que presenta la umbralización global, se recurre a la umbralización local en la que la clasificación de un píxel se realiza atendiendo a las propiedades locales del píxel (es decir, propiedades que dependen de su entorno) en vez de la propiedad global que es su intensidad luminosa.

El procedimiento que se sigue habitualmente a la hora de implantar una umbralización local es trabajar con el histograma de una subregión de la imagen global, existiendo básicamente dos posibilidades:

1. Dividir la imagen en un número de subregiones o ventanas cuadradas disjuntas tal y como se indica en la figura 3.17.

2. Definir una ventana cuadrada centrada en el píxel a clasificar. Este método implica una elevada carga computacional, por lo que puede ser impracticable en determinadas situaciones.

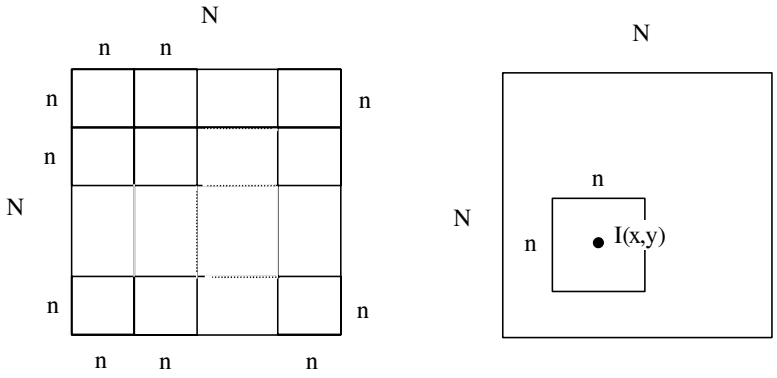


Figura 3.17: En la imagen de la izquierda se visualiza la división de la imagen global de dimensión Nx N en un conjunto de cuadrículas de dimensión nxn. En la figura de la derecha se calcula el histograma de los nxn píxeles centrado en cada píxel genérico de intensidad I(x,y).

Una vez establecidas las subregiones de la imagen, por cualquiera de los dos métodos anteriores, la umbralización se realizará siguiendo las mismas ideas que se expusieron para la umbralización global.

Selección del umbral basada en los píxeles de la frontera

Uno de los aspectos más importantes a la hora de seleccionar el umbral es la capacidad de identificar los modos existentes en el histograma. Esto es particularmente importante cuando se selecciona automáticamente el valor del umbral en situaciones en donde la característica de la imagen puede cambiar dentro de un amplio rango de distribuciones de intensidad. Es evidente que la probabilidad de seleccionar un

buen umbral aumenta si los modos del histograma son altos, estrechos, simétricos y están separados por profundos valles.

Un método para mejorar la apariencia del histograma es considerar únicamente los píxeles sobre o cercanos a los contornos entre los objetos y el fondo. Una mejora inmediata y obvia es que de esta forma se obtienen histogramas menos dependientes del tamaño relativo de los objetos frente al fondo.

El principal problema de este método es que se supone que el límite entre objetos y fondo es conocido a priori. Esta información no es disponible durante la segmentación ya que encontrar la división entre objetos y fondo es precisamente el objetivo que se pretende abordar. Sin embargo, mediante detectores de bordes es posible determinar si un píxel está sobre un borde.

En el esquema de la figura 3.18 se puede apreciar el procedimiento de segmentación basado en la extracción de bordes. A la imagen original $f(x, y)$ se le aplica un filtro de detección de bordes (mediante alguno de los métodos estudiados en el capítulo anterior) de forma que se obtiene una imagen filtrada $g(x, y)$ en la que los niveles de intensidad de los píxeles que pertenecen a los bordes han sido realzados, mientras que los niveles de intensidad de los píxeles que no pertenecen a los bordes quedan disminuidos.

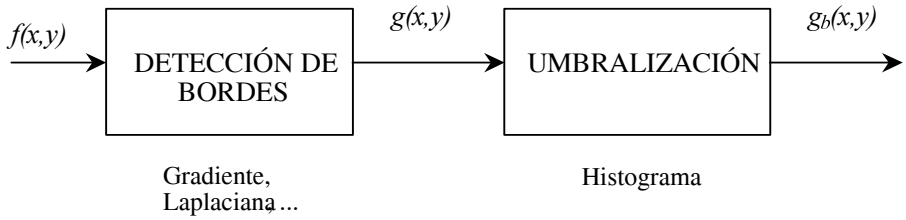


Figura 3.18: Esquema del proceso de segmentación de una imagen basado en umbralizar el histograma de la imagen original filtrada.

Con esta operación de filtrado la imagen resultante presenta un marcado carácter bimodal; es decir, en su histograma aparecen dos zonas diferenciadas: una formada

por los píxeles de los bordes y otra por los píxeles restantes tal y como se muestra en la figura 3.19.

La umbralización del histograma de la imagen filtrada producirá una imagen binaria $g_b(x, y)$ tal que:

$$g_b(x,y) = \begin{array}{ll} 1 & si\ (x,y) \in borde \\ 0 & si\ (x,y) \in resto \end{array} \tag{3.25}$$

Es inmediato aplicar las técnicas de umbralización del histograma que se estudiaron anteriormente para determinar la imagen binaria con la ventaja fundamental de que se trata de un problema bimodal o de dos clases y, además, en el que estas dos clases presentan distribuciones relativamente diferenciadas. No obstante, la realidad es que con frecuencia la imagen filtrada presenta problemas a la hora de umbralizar su histograma. Estos problemas se derivan de la naturaleza de la imagen original además de la calidad del filtrado que se haya aplicado.

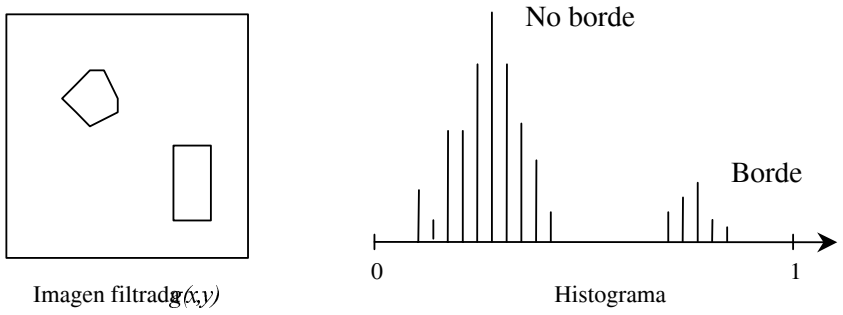


Figura 3.19: Histograma de los píxeles.

En consecuencia, con el fin de conseguir una imagen filtrada de gran calidad, desde el punto de vista de la detección de los bordes, es habitual combinar los efectos de un filtro basado en el gradiente y otro basado en la laplaciana.

El procedimiento se basa en formar una imagen con tres niveles o etiquetas para cada uno de los píxeles. Así, se analiza en primer lugar si un píxel pertenece o no a la categoría de borde para lo cual se umbraliza el histograma de la imagen filtrada con un operador de tipo gradiente. Los píxeles que se encuentran por debajo de un

umbral U, reciben la etiqueta O. El resto de los píxeles deberán pasar el siguiente test, esta vez aplicando el operador laplaciana. Si un píxel con el gradiente por encima de U tiene un valor positivo o igual a cero de su laplaciana, entonces se le aplica la etiqueta $+$, indicando que se trata de un píxel situado en una transición de nivel oscuro a claro.

Análogamente, si es un píxel con un gradiente por encima de U y con su laplaciana negativa se etiquetará con un símbolo $-$ puesto se trata de un píxel ubicado en una transición de nivel claro a oscuro.

Por tanto, la imagen resultante será:

$$S(x,y) = \begin{array}{ll} 0 & si\ f_G(x,y) < U \\ + & si\ f_G(x,y) \geq U,\ f_L(x,y) \geq 0 \\ - & si\ f_G(x,y) < U,\ f_L(x,y) < 0 \end{array} \tag{3.26}$$

Nótese cómo ahora no es necesario umbralizar la imagen de salida para obtener los bordes, puesto que éstos se calcularan directamente del análisis de las transiciones de los código $+$ y $-$. En la figura 3.20 se muestra la imagen resultante tras el proceso de etiquetado.

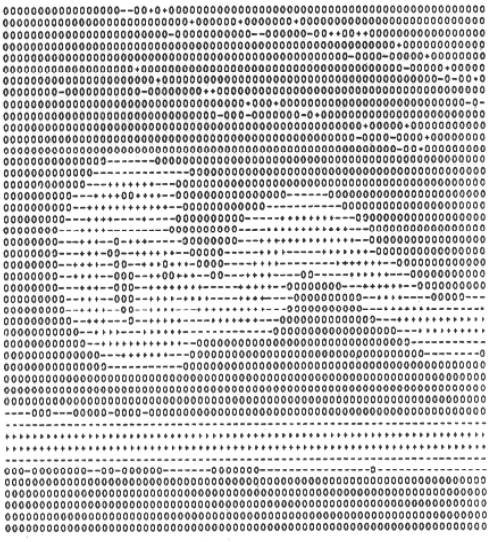


Figura 3.20: Imagen resultante cuando se utiliza la selección del umbral basada en los píxeles de la frontera (Gonzalez y Wood).

3.4 Segmentación basada en la agrupación de píxeles

En esta sección se describen técnicas que están basadas en la búsqueda de regiones directamente. Realmente las técnicas de agrupación de píxeles son equivalentes a las técnicas de agrupación de datos (*clustering*) caracterizadas por tratarse de técnicas no supervisadas, es decir, no existe un conocimiento previo de las clases de objetos existentes. La única información necesaria es la definición inicial de un vector X de características para los píxeles de la imagen a segmentar. Algunos de estos algoritmos, a lo sumo, precisan conocer también el número de clases existentes en la imagen.

Una vez establecido el vector de características, los procedimientos de agrupación de clases reciben como datos de entrada los objetos a clasificar, de modo que a partir de estos datos de entrada el algoritmo, sin supervisión de ningún tipo y de forma autónoma, agrupa estos vectores en clases. Por esta razón se les ha denominado también algoritmos de clasificación autoorganizada. La figura 3.21 pretende representar gráficamente la acción de las técnicas de agrupación de datos.

Figura 3.21: Representación simbólica de un algoritmo de agrupación.

Los algoritmos varían entre sí por el mayor o menor grado de reglas heurísticas que utilizan e, inversamente, por el nivel de procedimientos formales involucrados. Todos ellos se basan en el empleo sistemático de las distancias entre los vectores (los objetos a clasificar) así como entre los grupos que se van haciendo y deshaciendo a lo largo del proceso de actuación del algoritmo. Habitualmente se emplea la distancia

euclídea entre vectores:

$$d_E(X_i, X_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{3.27}$$

A continuación vamos a analizar los siguientes algoritmos presentados de menor a mayor complejidad descritos en [7] y [5]:

1. Crecimiento de regiones mediante la adición de píxeles.
2. Algoritmo de las distancias encadenadas.
3. Algoritmo max-min.
4. Algoritmo de K-medias.
5. Técnica de dividir y unir regiones.

3.4.1 Crecimiento de regiones mediante adición de píxeles

El crecimiento de regiones es un procedimiento que agrupa píxeles o subregiones en regiones más grandes. El método más simple es la adición de píxeles partiendo de un conjunto de puntos semillas a los que se van añadiendo píxeles vecinos que poseen propiedades similares (por ejemplo, el nivel de gris, textura, color,....). Si se usan n semillas, al final se podrá obtener una segmentación con un máximo de n regiones, además del fondo.

Veamos un ejemplo: considérese la figura 3.22 (a), donde los números dentro de cada celda representan valores de intensidad. Supóngase que se eligen como semillas las celdas $(3, 2)$ y $(3, 4)$. Al usar dos semillas se obtienen dos regiones: R_1 asociada con la semilla $(3, 2)$, y R_2 asociada con la semilla $(3, 4)$. La propiedad o predicado empleado para añadir un píxel a una región es que la diferencia en valor absoluto entre la intensidad de dicho píxel y la semilla sea menor que un umbral U. Si un píxel satisface esta propiedad para las dos semillas consideradas se asignará a la

región R_1. El resultado obtenido para $U = 3$ se muestra en la figura 3.22 (b), donde los píxeles que pertenecen a R_1 se han marcado como a, mientras que los que pertenecen a R_2 se han marcado con una b.

Obsérvese que, si se hubiera tomado como valor del umbral $U = 8$, el resultado dependería del orden de crecimiento de las regiones. Así si se deja crecer primero la región R_1 el resultado es una única región tal y como se muestra en la figura 3.22 (c). Es decir, este método presenta el inconveniente de que puede dar diversos resultados dependiendo de donde se sitúen las semillas y del umbral empleado para determinar si un píxel pertenece o no a una región.

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
0	1	5	6	5
0	1	5	6	5

(a)

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

(b)

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

(c)

Figura 3.22: Ejemplo de la técnica de crecimiento de regiones

Por tanto, esta técnica plantea como problemas fundamentales: la colocación de las semillas y la elección de las propiedades para la inclusión de píxeles en cada región. La colocación de las semillas se suele basar en la naturaleza del problema, y requiere algún tipo de información previa de las regiones a segmentar. Una solución general consiste en elegir píxeles cuyo nivel de gris correspondan a picos del histograma. Respecto a las propiedades que controlen la incorporación de nuevos píxeles pueden tener en cuenta diferentes características de la imagen (textura, niveles de gris, niveles de gris más localización).

Finalmente hay que comentar que una región deja de crecer cuando no existen más píxeles que satisfagan el criterio de inclusión en esa región. Durante la ejecución del algoritmo cabe la posibilidad que se llegue a un punto muerto donde queden píxeles sin clasificar pero las regiones no puedan crecer más por lo que es necesario volver a lanzar más semillas.

3.4.2 Algoritmo de las distancias encadenadas

Se trata de un algoritmo muy simple que no requiere ninguna información a priori de la distribución por clases de los objetos a clasificar. Aunque los resultados pueden no ser, para determinadas situaciones, los óptimos, es recomendable como procedimiento inicial para tantear la agrupación de los objetos.

Dados los vectores a clasificar (es decir los objetos) X_1, X_2, ..., X_p se escoge uno de ellos al azar, digamos X_i, y seguidamente se ordenan según la sucesión:

$$X_i(0),\ X_i(1),\ X_i(2)....X_i(k),\ X_i(k+1)....X_i(p-1)$$

en donde esta sucesión se ha formado de tal manera que el siguiente vector de la cadena es el más próximo al anterior. Es decir, $X_i(1)$ será el vector más próximo a $X_i(0)$, el $X_i(2)$ será el más cercano al $X_i(1)$, y así sucesivamente. Obsérvese que la sucesión formada depende del elemento inicial.

A continuación se calcula la sucesión de las distancias euclídeas relativas:

$$d_1,\ d_2,\ .....d_{k+1}.....d_{p-1}$$

siendo:

$$\begin{aligned} d_1 &= \|X_i(1) - X_i(0)\| \\ d_2 &= \|X_i(2) - X_i(1)\| \\ &..... \\ d_{k+1} &= \|X_i(k+1) - X_i(k)\| \\ d_{p-1} &= \|X_i(p-1) - X_i(p-2)\| \end{aligned} \tag{3.28}$$

A partir de su representación gráfica se puede detectar fácilmente una clase por el salto significativo en el valor de la correspondiente distancia euclídea (ver figura 3.23).

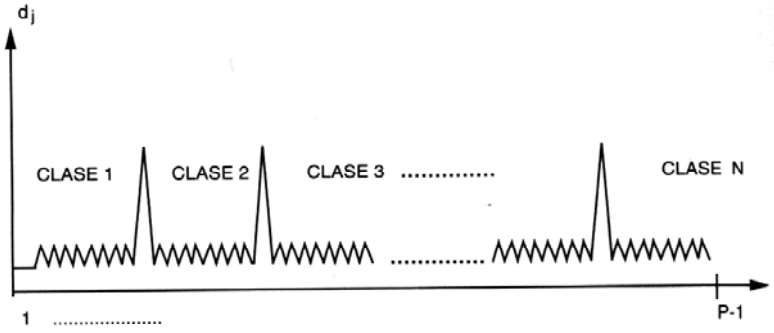


Figura 3.23: Representación idealizada del histograma distancia euclídea versus índice de la distancia.

El parámetro más delicado del algoritmo es el umbral de detección de una nueva clase. Un umbral excesivamente bajo dará lugar a clases ficticias, mientras que un umbral alto tenderá a agrupar en una misma clase objetos de diferentes clases.

En cuanto a la elección del primer elemento de la cadena, el algoritmo no se puede considerar excesivamente sensible, salvo para distribuciones especiales. En la figura 3.24 (a) se muestra una distribución problemática y en (b) el caso contrario.

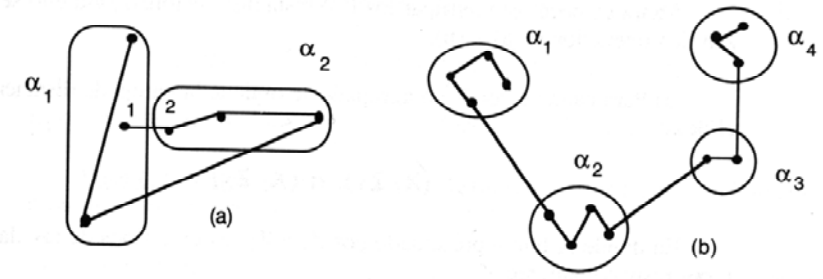


Figura 3.24: En (a) se representa una distribución de dos clases que pueden dar problemas si se escoge al azar como primer elemento el 1. La distribución en (b) no produce problema alguno(Darío Maravall).

3.4.3 Algoritmo de max-min

Es un algoritmo heurístico que emplea como único elemento formal la distancia euclídea. Tampoco requiere ninguna información a priori respecto al número de clases existentes.

Partiendo del conjunto de muestras o vectores a agrupar, los pasos del algoritmo son los siguientes:

- **Paso 1.** Se escoge al azar un elemento individual de los p disponibles, digamos X_i y se crea una primera clase α_1:

$$X_i \rightarrow \alpha_1$$

- **Paso 2.** Se calculan las distancias euclídeas de los $p-1$ vectores no agrupados y se toma la máxima distancia, de tal manera que el elemento implicado produce la segunda clase:

$$X_j / d\ (X_i, X_j)\ es\ máxima:\ X_j \rightarrow \alpha_2$$

- **Paso 3.** Se dispone ya de dos clases con sus correspondientes prototipos:

$$X_i \rightarrow \alpha_1\ ; \qquad X_j \rightarrow \alpha_2$$

ahora es necesario agrupar los $p-2$ restantes vectores, para ello se realizan dos operaciones:

1. Para cada vector X no agrupado se obtiene la pareja de distancias euclídeas:

$$d(X, Z_1),\ d(X, Z_2)$$

en donde se han representado por Z_1 y Z_2 los prototipos de las clases α_1 y α_2, respectivamente.

De cada $p-2$ parejas de distancias euclídeas previamente calculadas se toma la mínima distancia.

2. Del conjunto de las $p-2$ distancias mínimas obtenidas en el subpaso anterior se toma la distancia máxima, digamos d_{MAX}. Si esta distancia es superior a una determinada fracción de la distancia $d(Z_1, Z_2)$ entre los prototipos de las dos clases previamente formadas, entonces se crea una tercera clase. Es decir:

$$si\ d_{MAX} > d(Z_1, Z_2) * f \rightarrow se\ crea\ una\ clase\ \alpha_3$$

con $0 < f < 1$. El prototipo de α_3 es el elemento correspondiente a la distancia máxima d_{MAX}.

- **Paso 4.** Quedan $p-3$ elementos por clasificar. El proceso ahora es en todo similar; es decir, se calculan las tres distancias euclídeas de cada uno de los $p-3$ elementos sin clasificar a los prototipos α_1, α_2 y α_3 y se toma únicamente la mínima distancia de cada terna. A continuación, de entre las $p-3$ distancias mínimas se toma la máxima, de manera que si supera una fracción de la distancia media entre los prototipos de las clases formadas, se crea entonces una nueva clase α_4. Esto es:

$$\begin{gathered} d(X, Z_1), d(X, Z_2), d(X, Z_3) \rightarrow MINIMA : d_{MIN} \\ \{d^1_{MIN}, d^2_{MIN} .... d^{p-3}_{MIN}\} \rightarrow MAXIMA : d_{MAX} \end{gathered} \tag{3.29}$$

$$si\ d_{MAX} > f * \frac{1}{3}\{d(Z_1, Z_2) + d(Z_1, Z_3) + d(Z_2, Z_3)\} \rightarrow se\ crea\ \alpha_4$$

con $0 < f < 1$.

- **Paso 5.** Se continúa este proceso hasta que no se cumpla la condición:

$$d_{MAX} > f * distancia\ media\ entre\ clases \tag{3.30}$$

momento a partir del cual ya no es posible crear más clases.

- **Paso 6.** Los elementos que queden por agrupar se asignan a la clase cuyo prototipo esté más cerca.

Este algoritmo tiene un inconveniente, ya que es sensible al coeficiente f. Una buena elección de f es fundamental para el correcto funcionamiento de este procedimiento. No obstante, al tratarse de una técnica iterativa, es posible ensayar sucesivos valores hasta que se obtenga una agrupación final correcta.

La explicación intuitiva de cómo opera este algoritmo se basa en la combinación heurística de distancias euclídeas mínimas y máximas. En efecto, como el algoritmo básicamente está orientado a crear (o no) una nueva clase en cada paso iterativo, se calculan las distancias mínimas a las clases ya existentes, de tal manera que la máxima distancia de entre ellas se compara con la distancia media entre las clases ya formadas para verificar si es posible crear una nueva clase. En definitiva, se está probando la viabilidad de formar una nueva clase con un elemento lo suficientemente separado de las clases ya existentes.

3.4.4 Algoritmo k-medias

Este algoritmo hace referencia a que se conoce a priori que existen k clases o patrones es decir el número de clases existentes. Es un algoritmo sencillo pero muy eficiente siempre que el número de clases se conozca a priori con exactitud. Debido precisamente a su sencillez ha sido ampliamente utilizado.

Partiendo de un conjunto de objetos a clasificar X_1, $X_2, ...X_p$ este algoritmo realiza las siguientes operaciones:

- **Paso 1.** Establecido previamente el número exacto de clases existentes, digamos k, se escogen al azar entre los elementos a agrupar k vectores, de forma que van a constituir los centroides de las k clases. Es decir:

$$\alpha_1 : Z_1(1);\ \alpha_2 : Z_2(1); .....\alpha_k : Z_k(1)$$

  en donde se ha introducido entre paréntesis el índice iterativo de este algoritmo.

- **Paso 2.** Como se trata de un proceso recursivo con un contador n, en la iteración genérica n se distribuyen todas las muestras $\{X\}_{1\leq j\leq p}$ entre las k clases, de acuerdo con la siguiente regla:

$$X \in \alpha_j(n) \; si \; \|X - Z_j(n)\| < \|X - Z_i(n)\| \\ \forall \; i = 1, 2....k \; con \; i \neq j$$

en donde se han indexado las clases y sus correspondiente centroides.

- **Paso 3.** Una vez redistribuidos los elementos a agrupar entre las diferentes clases, es preciso recalcular o actualizar los centroides de las clases. El objetivo en el cálculo de los nuevos centroides es minimizar el índice de rendimiento siguiente.

$$J_i = \sum_{X \in \alpha_i(n)} \|X - Z_i(n)\|^2; \; i = 1, 2....k \tag{3.31}$$

Este índice se minimiza utilizando la media aritmética de $\alpha_i(n)$:

$$Z_i(n+1) = \frac{1}{N_i(n)} \sum_{X \in \alpha_i(n)} X; \; i = 1, 2....k \tag{3.32}$$

siendo $N_i(n)$ el número de elementos de la clase α_i en la iteracción n.

- **Paso 4.** se comprueba si el algoritmo ha alcanzado una posición estable. Es decir, si se cumple:

$$Z_i(n+1) = Z_i(n) \qquad \forall \; i = 1, 2....k \tag{3.33}$$

Si se cumple, el algoritmo finaliza. En caso contrario, salta al paso 2.

El algoritmo k-medias es simple y muy eficiente si se conoce con exactitud el número de clases. Un valor de k superior al número real de clases dará lugar a clases ficticias, mientras que un k inferior producirá menos clases de las reales. Una forma de detectar una mala elección del valor de k es analizar las dispersiones estadísticas de las clases formales. Cuando estas dispersiones sean sensiblemente diferentes entre

sí, siendo algunas de ellas muy elevadas respecto a las demás, se puede sospechar que se ha manejado un valor de k bajo. Análogamente, si algunas distancias interclases son muy pequeñas respecto a las demás distancias interclases, entonces es lógico plantearse que el parámetro k introducido es alto.

3.4.5 División y unión de regiones

En primer lugar se introducen algunos conceptos necesarios. Sea R una región que representa a la imagen completa. La segmentación de la imagen puede entenderse como el proceso que divide la región R en n subregiones, R_1, R_2, ..., R_n, tal que:

(a) $\bigcup_{i=1}^{n} R_i = R$

(b) R_i *con* $i = 1, 2, ..., n$ *es una región conectada*

(c) $R_i \bigcap R_j = \emptyset \qquad \forall i, j \quad i \neq j$

(d) $P(R_i) = TRUE \quad para\ i = 1, 2, ..., n$

(e) $P(R_i \bigcup R_j) = FALSE \quad para\ i \neq j$

Donde $P(R_i)$ es un predicado lógico definido sobre los puntos del conjunto R_i, basado en alguna medida de similitud, y $\emptyset$ es el conjunto vacío.

La condición (a) indica que la unión de todas las regiones obtenidas tras la segmentación debe ser la imagen completa. La condición (b) requiere que los puntos de una región estén conectados. La condición (c) indica que las regiones deben ser disjuntas. La condición (d) indica que los píxeles de una región segmentada deben satisfacer determinadas propiedades. Finalmente la condición (e) indica que las regiones R_i y R_j son distintas según el criterio del predicado P.

A éstas hay que añadir las siguientes definiciones:

- La región R_k se considera formada por un conjunto de puntos con las siguientes propiedades:

- x_i en una región R está conectado a x_j si hay una secuencia $\{x_i, ..., x_j\}$ tal que dos puntos intermedios x_k y x_{k+1} están conectados y todos ellos pertenecen a R.
- R es una región conectada si el conjunto de puntos x en R cumple la propiedad de que cada par de puntos está conectado.

Esta técnica se basa en la subdivisión inicial de la imagen en un conjunto de regiones arbitrarias y disjuntas. A partir de este conjunto, se comienza un proceso iterativo unión y/o división de dichas regiones, sujeto a las condiciones establecidas en la introducción de esta sección.

Más concretamente, la técnica de dividir y unir regiones consiste en:

Sea un conjunto de regiones R_k, $k = 1, ..., m$, si no se satisface la condición (d) para algún k significa que esa región debería ser dividida en regiones más pequeñas. Si es la condición (e) la que no se satisface para algún i y j, entonces las regiones i y j deberían unirse para formar una única región.

Una forma de trabajar con este algoritmo consiste en organizar los píxeles de la imagen en una estructura de malla piramidal de regiones. En esta estructura de malla, las regiones son organizadas en grupos de cuatro. Cualquier región podría dividirse en cuatro subregiones y viceversa cuatro regiones pueden unirse para formar una única región más grande. Esta estructura es la utilizada en el siguiente algoritmo:

1. Sea R la región inicial, constituida por la imagen completa.
2. Seleccionar un predicado P. Por ejemplo, un predicado válido podría ser la desviación típica la cual puede ser definida como:

$$\sigma^2 = \frac{1}{n} \sum (f(x, y) - \mu)^2 \tag{3.34}$$

donde μ es la media dada por:

$$\mu = \frac{1}{n} \sum f(x, y) \tag{3.35}$$

3. Para toda región R_i, tal que, $P(R_i) = FALSE$: Subdividir R_i en cuatro cuadrantes disjuntos. En el ejemplo propuesto de la desviación típica $P(R_i) = FALSE$ es equivalente a que la desviación típica sea mayor a un valor dado lo que indica una gran desviación en los valores de gris de la imagen.

4. Fusionar cualquier par de regiones adyacentes R_j y R_k, para las que se verifique $P(R_j \cup R_k) = TRUE$.

5. Si existen más regiones para fusionar o dividir entonces volver al paso 3, sino, parar.

Para la representación de las sucesivas subdivisiones se puede emplear un árbol cuaternario tal y como se ilustra en la figura 3.25. Precisamente este tipo de representación será estudiada en detalle en el siguiente capítulo.

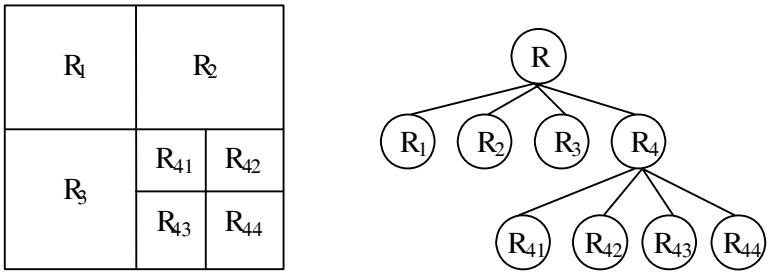


Figura 3.25: Ejemplo del método división y unión de regiones

Software

Ver capítulo 2.

Conocimientos nuevos adquiridos

El alumno a lo largo del tema ha estudiado algunas de las técnicas que se aplican durante la etapa de segmentación. No sólo es necesario que haya aprendido cada una de estas técnicas sino que también debe saber cuando resultará más conveniente el empleo de una u otra dependiendo de la imagen de partida y del tipo de objetos presentes.

Bibliografía complementaria

- Capítulo 6 de J. González: "Visión por computador". Paraninfo, 1999. [5].
- Capítulos 12 de D. Maravall: "Reconocimiento de formas y visión artificial". Rama 1993. [7]
- Capítulo 7 de A. De la Escalera: "Visión por computador. Fundamentos y métodos". Prentice Hall, 2001. [4].
- Capítulo 6 de R. J. Schalkoff: "Digital Image Processing and Computer Vision". John Wiley & Sons, inc., 1989. [9]
- Capítulo 5 de M. Sonka, V. Hlavac y R. Boyle: "Imagen Processing, Analysis and Machine Vision". Chapman & Hall Computing, 1993. [10].
- Capítulos 4 y 5 de D. H. Ballard y C. M. Brown: " Computer Vision". Prentice-Hall, 1982. [1].
- Capítulo 7 de R. C. González y P. Wintz: "Digital Image Processing". Addison-Wesley Publishing Company, 1987. [6].
- Capítulo 4 de E. R. Davies:"Machine Vision. Theory. Algorithms. Practicalities". Academic Press, 1990". [3].

Actividades

- Realizar y comprender los ejercicios resueltos suministrados por el equipo docente.
- Desarrollar utilizando cualquiera de las herramientas software los distintos algoritmos estudiados a lo largo de la asignatura. Ello permitirá afianzar conocimientos y detectar la dificultad del proceso en algunos casos. Para ello se suministrará al alumno, junto con el software, una biblioteca de imágenes con las que podrá trabajar.

Autoevaluación

Dada una imagen, el alumno debe de saber qué método aplicar para detectar los objetos presentes en la imagen.

También puede realizar las cuestiones y problemas de exámenes de los cursos anteriores.

Problemas resueltos

PROBLEMA 1

Implementar, mediante pseudocódigo, el algoritmo para la segmentación basada en la detección de fronteras utilizando el análisis local para la detección de bordes. Considérese una imagen de $N \times M$ la cual representa a un único objeto de contorno cerrado.

¿Qué se pretende con este problema? Con este problema se intenta introducir al alumno en la programación de los métodos para la segmentación de la imagen. Es conveniente para la implementación del algoritmo disponer de las funciones que determinen la imagen resultante tras aplicar cualquiera de los operadores estudiados en el capítulo anterior para la detección de los bordes de los objetos.

Solución

N: número de filas en la imagen.

M: número de columnas en la imagen.

contorno: variable para almacenar las coordenadas (x,y) de los bordes.

tabla: variable para etiquetar los píxeles de la imagen:

píxel = 1 si ya ha sido considerado.

píxel = 0 si no ha sido considerado.

píxel = 2 primer píxel.

vecindad: vecindad considerada durante el algoritmo.

U: valor del umbral para la diferencia de magnitud.

A: valor del umbral para la diferencia de ángulo.

T: umbral para la selección del primer borde.

ind: contabiliza el número de píxeles del contorno.

FIN: variable para detectar el fin del proceso de búsqueda de nuevos bordes.

candidato: variable para almacenar los posibles píxeles del contorno.

;Detección de los bordes empleando cualquiera de los operadores estudiados en el capítulo anterior (Prewitt, Sobel, Roberts,...):

I_x = imagen resultante en la dirección x.

I_y = imagen resultante en la dirección y.

I = imagen resultante de la combinación de gradientes.

; Determinación de la dirección del gradiente asociada a cada píxel:

for x igual a 1 hasta N

for y igual a 1 hasta M

if $I_x(x,y)$ *igual a 0 entonces*

$\alpha(x,y) \leftarrow 90^o$

else

$\alpha(x,y) \leftarrow arctag\left(\frac{I_y(x,y)}{I_x(x,y)}\right)\frac{180^o}{\pi}$

end

end

end

; Inicialización de variables:

FIN$\leftarrow$ 0, elementos de la matriz tabla = 0, vecindad = 1.

; Detección del primer borde:

x ← 1

y ← 1

mientras I(x,y) menor a T and (x menor a N | y menor a M)

; Avanzar al siguiente píxel en la imagen

end

if x igual a N and y igual a M

;No se han detectado objetos en la imagen.

FIN ← 1

else

ind ← 1

contorno ← (x,y)

tabla(x,y) ← 2

end

; Seguimiento del contorno:

candidato ← iniciar

mientras FIN igual a 0

(x,y) ← último punto asociado al contorno

; Búsqueda del siguiente borde perteneciente a la vecindad del anterior:

for i = x-vecindad hasta x+vecindad

for j = y-vecindad hasta y+vecindad

if x ≠i and y≠j entonces

if tabla(i,j) igual a 0

if $|I(x,y) - I(i,j)| < U$ *and* $|\alpha(x,y) - \alpha(i,j)| < A$

candidato← *(i,j)*

tabla(i,j) ← *1*

end

end

if tabla(i,j) = 2 and (x,y) no pertenece a vecindad 1^{er} *borde*

; Ultimo borde: se ha regresado a la posición inicial:

FIN ← *1*

contorno← *(i,j)*

ind ← *ind + 1*

end

end

end

end

; Selección del candidato

if FIN igual a 0

max ← *primer elemento de candidato*

for k =igual a 2 hasta el último elemento de candidato

if elemento actual de candidato mayor a max

max ← *elemento actual de candidato*

end

end

;Almacenamos el nuevo borde

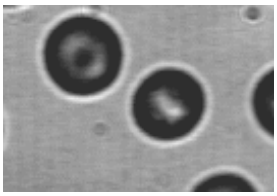
ind ← *ind* + *1*

contorno ← *max*

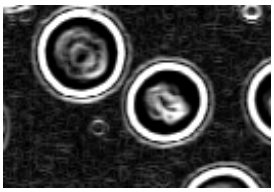
end

end

En la figura 3.26 se muestra un ejemplo de aplicación del algoritmo anterior.



(a) Imagen original



(b) Detección de bordes con el operador Sobel

(c) Segmentación de uno de uno de los objetos

Figura 3.26: Aplicación del algoritmo

Se recomienda al alumno las siguientes actividades:

1. Implementar el algoritmo descrito anteriormente en *scilab*. Estudiar, con diferentes imágenes constituidas por un único objeto, el efecto de la variación del valor de los umbrales considerados (U, A, T) sobre la detetección del contorno.

2. Modificar adecuadamente el algoritmo para aumentar la vecindad únicamente cuando sea necesario, es decir, cuando no se encuentre ningún candidato entre los vecinos correspondientes.

3. Modificar el algoritmo para conseguir detectar contornos abiertos.

4. Ampliar el algoritmo para detectar todos los contornos de una imagen caracterizada por tener más de un objeto.

PROBLEMA 2

Aplicar la transformación de Hough para la detección de elipses orientadas de manera que el eje principal sea paralelo al eje X. Suponer que se dispone de un conjunto de características locales tales como el conjunto de puntos con sus tangentes asociadas.

¿Qué se pretende con este problema? Este problema muestra cómo se obtiene la transformación Hough de una curva definida mediante su ecuación paramétrica, realizando la simplificación relativa al conocimiento de la tangente en cada punto. Recuérdese que la transformada Hough es un método para la obtención del contorno de un objeto en la imagen aplicable siempre que se tenga un conocimiento aproximado de la localización del limite y su forma pueda ser descrita mediante una curva paramétrica.

Solución

Las elipses cuyo eje principal es paralelo al eje X pueden especificarse mediante cuatro parámetros (x_0, y_0, a, b) tal y como se muestra en la ecuación:

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1 \tag{3.36}$$

Sin embargo, utilizando esta ecuación junto con su derivada y sustituyendo el valor del gradiente de manera análoga a lo expuesto en la parte teórica, es posible reducir los parámetros a 2. La ecuación de su derivada viene dada por:

$$\frac{(x-x_0)}{a^2} + \frac{(y-y_0)}{b^2} = 0 \tag{3.37}$$

donde $\frac{dy}{dx} = \tan\,\phi(x)$.

Operando en estas dos ecuaciones se obtiene la transformada de Hough:

$$x = x_0 \pm \frac{a}{(1+\frac{b^2}{a^2}\tan^2\phi)^{\frac{1}{2}}} \tag{3.38}$$

$$y = y_0 \pm \frac{b}{(1+\frac{a^2}{b^2}\tan^2\phi)^{\frac{1}{2}}} \tag{3.39}$$

Supongamos que consideramos cada vez parejas de puntos (bordes). Esto introduce dos ecuaciones por cada uno de los puntos, esto es:

$$\frac{(x_1 - x_0)^2}{a^2} + \frac{(y_1 - y_0)^2}{b^2} = 1 \quad (3.40)$$

$$\frac{(x_2 - x_0)^2}{a^2} + \frac{(y_2 - y_0)^2}{b^2} = 1 \quad (3.41)$$

$$\frac{(x_1 - x_0)}{a^2} + \frac{(y_1 - y_0)}{b^2}\frac{dy}{dx} = 0 \quad (3.42)$$

$$\frac{(x_2 - x_0)}{a^2} + \frac{(y_2 - y_0)}{b^2}\frac{dy}{dx} = 0 \quad (3.43)$$

de donde es posible el cálculo exacto de los parámetros desconocidos sabiendo que $\frac{dy}{dx} = \tan\phi$.

PROBLEMA 3

Describir el algoritmo de segmentación basado en la transformación de Hough para la detección de:

1. Rectas en la imagen.
2. Objetos con cualquier contorno.

¿Qué se pretende con este problema? Con este problema se intenta introducir al alumno en la programación de las técnicas de segmentación basadas en la transformación de Hough.

Solución

1. *Detección de rectas*

   - Discretizar el espacio de parámetros, estableciendo valores máximos y mínimos de τ y θ, así como el número total de valores de τ y θ.
   - Generar el acumulador $A(\tau, \theta)$; poner todos los valores a 0.
   - Para todos los bordes (x_i, y_i):
     - calcular la dirección del vector gradiente θ.

  - obtener la τ de la ecuación $x_i \cdot cos\ \theta + y_i \cdot sen\ \theta = \tau$
  - incrementar $A(\tau, \theta)$
- Para todas las celdas en el acumulador
  - buscar los valores máximos del acumulador
  - las coordenadas (τ, θ) dan la ecuación de la recta en la imagen.

2. *Detección de contornos generales*

- Construir la R-tabla a partir del objeto prototipo.
- Para todos los puntos de la frontera:
  - calcular la orientación α (dirección del gradiente +90).
  - calcular r y β.
  - añadir un (r, β) en la R-tabla indexado por α.
- Discretizar el espacio de parámetros, estableciendo valores máximos y mínimos, así como el número total de valores de x_{ref}, y_{ref} y Ω.
- Generar el acumulador $A(x_{ref}, y_{ref}, \Omega)$; poner todos los valores a 0.
- Para todos los puntos del borde (x_i, y_i)
  - calcular la orientación α (dirección del gradiente +90)
  - para cada orientación posible Ω
    - para cada par (r, β) indexado por $\alpha - \Omega$ en la R-tabla
    - evaluar:

$$x_{ref} = x + r \cdot cos\ (\beta + \Omega)$$
$$y_{ref} = y + r \cdot sen\ (\beta + \Omega)$$

    - incrementar $A(x_{ref}, y_{ref}, \Omega)$
- Para todas las celdas en el acumulador
  - buscar los valores máximos del acumulador

– las coordenadas x_{ref}, y_{ref} y Ω dan la posición y orientación del objeto en la imagen

Se recomienda al alumno que implemente los algoritmos anteriores en *scilab* probando su eficacia con diferentes imágenes.

PROBLEMA 4

Supongamos que la imagen considerada está formada por dos clases α_1 y α_2, de las que se dispone de un conjunto de cinco muestras por clase $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$:

$$\alpha_1 \ : \ \left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\}$$

$$\alpha_2 \ : \ \left\{ \begin{pmatrix} 8 \\ 10 \end{pmatrix}, \begin{pmatrix} 9 \\ 8 \end{pmatrix}, \begin{pmatrix} 9 \\ 9 \end{pmatrix}, \begin{pmatrix} 8 \\ 9 \end{pmatrix}, \begin{pmatrix} 7 \\ 9 \end{pmatrix} \right\}$$

Obtener un reconocedor estadístico para la obtención de las regiones de la imagen.

¿Qué se pretende con este problema? Con este problema se pretende mostrar al alumno con un ejemplo muy sencillo como se aplican las técnicas bayesianas para la obtención de un reconocedor capaz de clasificar un vector de características X relativo a la imagen. Recuérdese que las técnicas bayesianas se aplicaban en la umbralización basada en las técnicas de reconocimiento de formas cuando se conoce de antemano el número de clases existentes en la imagen.

Solución

El primer paso será estimar los vectores media y las matrices de covarianza:

- *Media*

$$\hat{m}_1 = \frac{1}{5} \sum_{j=1}^{5} X_{1,j} = \frac{1}{5} \begin{pmatrix} 11 \\ 10 \end{pmatrix} = \begin{pmatrix} 2'2 \\ 2 \end{pmatrix}$$

$$\hat{m}_2 = \frac{1}{5} \sum_{j=1}^{5} X_{2,j} = \frac{1}{5} \begin{pmatrix} 41 \\ 45 \end{pmatrix} = \begin{pmatrix} 8'2 \\ 9 \end{pmatrix}$$

- *Covarianza*

$$\hat{C}_1 = \frac{1}{5}\sum_{j=1}^{5} X_{1,j}\, X_{1,j}^T - \hat{m}_1 \hat{m}_1^T =$$

$$= \frac{1}{5}\left[\begin{pmatrix} 1 \\ 2 \end{pmatrix}\begin{pmatrix} 1 & 2 \end{pmatrix} + \begin{pmatrix} 2 \\ 2 \end{pmatrix}\begin{pmatrix} 2 & 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix}\begin{pmatrix} 3 & 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 3 \end{pmatrix}\begin{pmatrix} 2 & 3 \end{pmatrix} + \begin{pmatrix} 3 \\ 2 \end{pmatrix}\begin{pmatrix} 3 & 2 \end{pmatrix}\right] - \frac{1}{25}\begin{pmatrix} 11 \\ 10 \end{pmatrix}\begin{pmatrix} 11 & 10 \end{pmatrix}$$

$$= \frac{1}{25}\begin{pmatrix} 14 & -5 \\ -5 & 10 \end{pmatrix}$$

$$\hat{C}_2 = \frac{1}{5}\sum_{j=1}^{5} X_{2,j}\, X_{2,j}^T - \hat{m}_2 \hat{m}_2^T =$$

$$= \frac{1}{5}\left[\begin{pmatrix} 8 \\ 10 \end{pmatrix}\begin{pmatrix} 8 & 10 \end{pmatrix} + \begin{pmatrix} 9 \\ 8 \end{pmatrix}\begin{pmatrix} 9 & 8 \end{pmatrix} + \begin{pmatrix} 9 \\ 9 \end{pmatrix}\begin{pmatrix} 9 & 9 \end{pmatrix} + \begin{pmatrix} 8 \\ 9 \end{pmatrix}\begin{pmatrix} 8 & 9 \end{pmatrix} + \begin{pmatrix} 7 \\ 9 \end{pmatrix}\begin{pmatrix} 7 & 9 \end{pmatrix}\right] - \frac{1}{25}\begin{pmatrix} 41 \\ 45 \end{pmatrix}\begin{pmatrix} 41 & 45 \end{pmatrix}$$

$$= \frac{1}{25}\begin{pmatrix} 14 & -5 \\ -5 & 10 \end{pmatrix}$$

Al ser las dos matrices iguales, las funciones discriminantes son lineales y de la forma:

$$\begin{aligned} fd_1(X) &= X^T\, C^{-1}\, m_1 - \frac{1}{2}\, m_1^T\, C^{-1}\, m_1 \\ fd_2(X) &= X^T\, C^{-1}\, m_2 - \frac{1}{2}\, m_2^T\, C^{-1}\, m_2 \end{aligned}$$

Sustituyendo valores:

$$\begin{aligned} fd_1(X) &= \frac{32}{115}\, X_1 + \frac{39}{115}\, X_2 - 0'65 \\ fd_2(X) &= \frac{127}{115}\, X_1 + \frac{167}{115}\, X_2 - 0'11 \end{aligned}$$

La clasificación de un vector X será:

$$\begin{aligned} X &\in \alpha_1 \quad si\ fd_1(X) > fd_2(X) \longleftrightarrow p(X\ /\ \alpha_1) > p(X\ /\ \alpha_2) \\ X &\in \alpha_2 \quad si\ fd_1(X) < fd_2(X) \longleftrightarrow p(X\ /\ \alpha_1) < p(X\ /\ \alpha_2) \end{aligned}$$

en donde se ha supuesto que $p(\alpha_1) = p(\alpha_2) = 0'5$.

El lugar geométrico de los puntos situados en la frontera entre ambas clases resulta ser:

$$fd(X) = fd_1(X) - fd_2(X) \equiv \frac{X_1}{12} + \frac{X_2}{9'32} = 1$$

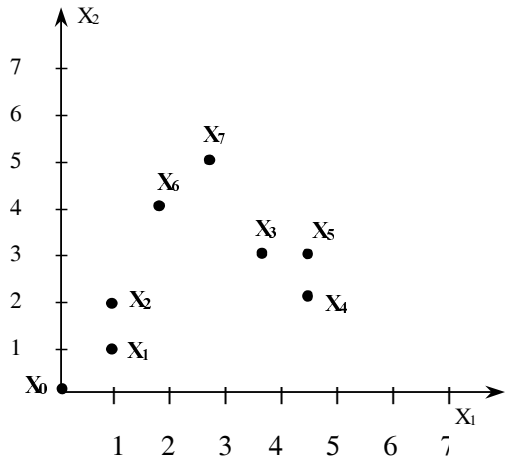
Por tanto, la clasificación en relación a la función discriminante conjunta será:

$$X \in \alpha_1 \quad si\ fd(X) > 0$$
$$X \in \alpha_2 \quad si\ fd(X) < 0$$

PROBLEMA 5

Sea una imagen en la que no existe conocimiento previo de las clases existentes y de la que se dispone el vector de características definido por:

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} \begin{pmatrix} 3 \\ 5 \end{pmatrix} \begin{pmatrix} 4 \\ 6 \end{pmatrix} \begin{pmatrix} 5 \\ 3 \end{pmatrix} \begin{pmatrix} 6 \\ 3 \end{pmatrix} \begin{pmatrix} 6 \\ 4 \end{pmatrix} \right\}$$



Aplicar las siguientes técnicas de segmentación basadas en el crecimiento de regiones:

1. Algoritmo de max-min.

2. Algoritmo de las distancias encadenadas.

¿Qué se pretende con este problema? En este problema se analiza con un ejemplo dos de los métodos expuestos en la parte teórica englobados en las técnicas de segmentación basadas en el crecimiento de regiones.

Solución

$$\underset{X_1}{\left\{\begin{pmatrix}1\\1\end{pmatrix}\right.}\ \underset{X_2}{\begin{pmatrix}1\\3\end{pmatrix}}\ \underset{X_3}{\begin{pmatrix}3\\5\end{pmatrix}}\ \underset{X_4}{\begin{pmatrix}4\\6\end{pmatrix}}\ \underset{X_5}{\begin{pmatrix}5\\3\end{pmatrix}}\ \underset{X_6}{\begin{pmatrix}6\\3\end{pmatrix}}\ \underset{X_7}{\left.\begin{pmatrix}6\\4\end{pmatrix}\right\}}$$

- *Algoritmo de max-min:*

Siguiendo los pasos del algoritmo descritos en la parte teórica de este capítulo se obtiene:

1. Supongamos que tomamos al azar $X_4 = \begin{pmatrix}4\\6\end{pmatrix}$:

$$X_4 \rightarrow \alpha_1(Z_1)$$

2. Calculamos las distancias euclídeas de los $p - 1 = 7 - 1 = 6$ vectores no agrupados:

$$d(X_4, X_1) = \sqrt{3^2 + 5^2} = 5'8$$
$$d(X_4, X_2) = \sqrt{3^2 + 3^2} = 4'2$$
$$d(X_4, X_3) = 1'4$$
$$d(X_4, X_5) = 3'1$$
$$d(X_4, X_6) = 3'6$$
$$d(X_4, X_7) = 2'8$$

La distancia máxima es $d(X_4, X_1)$, por lo que la segunda clase será:

$$X_1 \rightarrow \alpha_2(Z_2)$$

3. Calculamos la pareja de distancias euclídeas para cada vector X no agrupado. Los valores obtenidos se muestran en la siguiente tabla, en donde se ha señalado la distancia mínima de cada una de estas parejas.

$$\begin{array}{ll} d(Z_2, X_2) = \mathbf{2'23} & d(Z_1, X_2) = 4'2 \\ d(Z_2, X_3) = 4'47 & d(Z_1, X_3) = \mathbf{1'4} \\ d(Z_2, X_5) = 4'47 & d(Z_1, X_5) = \mathbf{3'1} \\ d(Z_2, X_6) = 5'38 & d(Z_1, X_6) = \mathbf{3'6} \\ d(Z_2, X_7) = 5'8 & d(Z_1, X_7) = \mathbf{2'8} \end{array}$$

Del conjunto de distancias mínimas obtenidas anteriormente se selecciona el valor máximo. Esto es:

$$max\left\{min\left\{d(X_j, Z_i)\right\}\right\} = d(Z_1, X_6) = 3'6$$

Si $d(Z_1, X_6) > f \cdot d(Z_1, Z_2)$ ($d(Z_1, Z_2)$ es la distancia media entre clases, entonces creamos una nueva clase. Así, tomando $f = 0'6$ y sabiendo que $d(Z_1, Z_2) = 5'8$ tenemos:

$$d(Z_1,X_6) = 3'6 > 0'6 \cdot 5'8 \Longrightarrow X_6 \rightarrow \alpha_3(Z_3)$$

Repetimos el proceso hasta que no sea posible crear más clases.

Distancias euclídeas:

$$\begin{array}{lll} d(Z_2, X_2) = \mathbf{2'23} & d(Z_1, X_2) = 4'2 & d(Z_3, X_2) = 5 \\ d(Z_2, X_3) = 4'47 & d(Z_1, X_3) = \mathbf{1'4} & d(Z_3, X_3) = 3'6 \\ d(Z_2, X_5) = 4'47 & d(Z_1, X_5) = 3'1 & d(Z_3, X_5) = \mathbf{1} \\ d(Z_2, X_7) = 5'8 & d(Z_1, X_7) = 2'8 & d(Z_3, X_7) = \mathbf{1} \end{array}$$

$$max\left\{min\left\{d(X_j, Z_i)\right\}\right\} = d(Z_2, X_2) = 2'23$$

$$\text{distancia media entre clases} = \begin{array}{l} d(Z_1, Z_2) = 5'8 \\ d(Z_1, Z_3) = 3'6 \\ d(Z_2, Z_3) = 5'38 \end{array} = \frac{5'8 + 3'6 + 5'38}{3} = 4'926$$

$$d(Z_2, X_2) = 2'23 > 0'6.4'926 \Rightarrow \textit{No existen más clases}$$

Finalmente agrupamos el resto de los elementos a su clase más cercana:

$$X_2 \in Z_2$$
$$X_3 \in Z_1$$
$$X_5 \in Z_3$$
$$X_7 \in Z_3$$

- *Algoritmo de las distancias encadenadas*

  Se elige uno de los elementos al azar, por ejemplo X_4, y se ordenan de forma que el siguiente elemento de la cadena resultante sea el más cercano:

  $$X_4, X_3, X_2, X_1, X_5, X_6, X_7$$

  Seguidamente se calcula la sucesión de las distancias euclídeas, denominadas d_1, d_2, d_3, d_4, d_5 y d_6, siendo:

  $$d_1 = d(X_4, X_3) = 1'4$$
  $$d_2 = d(X_3, X_2) = 2'8$$
  $$d_3 = d(X_2, X_1) = 2$$
  $$d_4 = d(X_1, X_5) = 4'47$$
  $$d_5 = d(X_5, X_6) = 1$$
  $$d_6 = d(X_6, X_7) = 1$$

  Observando la representación gráfica de las distancias euclídeas es posible detectar fácilmente una clase por el salto significativo en el valor de la correspondiente distancia euclídea. Si se elige como umbral el valor de $2'5$ se distinguen claramente tres clases, tal y como se muestra en la figura 3.27.

  Finalmente, el procedimiento para la asociación de los elementos restantes es semejante al método anterior, esto es, se agrupan a su clase más cercana.

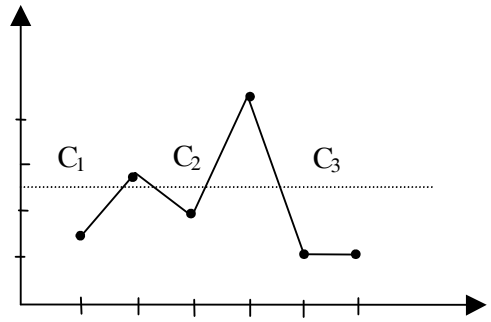


Figura 3.27: Representación gráfica de las distancias euclídeas.

Capítulo 4

Descripción de objetos

Introducción y orientaciones para el estudio

Una vez segmentada la imagen, el siguiente paso consiste en extraer las características discriminantes de los objetos que permitan su posterior reconocimiento. Para ello, en primer lugar, es necesario representar de alguna manera los objetos obtenidos en la segmentación. Se entiende por representación de un objeto al proceso que transforma los píxeles que lo integran a un formato más compacto y de un nivel superior.

La primera parte de esta sección se dedica a describir diferentes tipos de representaciones bidimensionales comentando la posibilidad de conseguir que sean invariantes a translaciones, rotaciones y homotecias. En la segunda parte se muestra al alumno cómo calcular diferentes tipos de características que permitan diferenciar unos objetos de otros. Hay que comentar que la representación de objetos tridimensionales así como el estudio de técnicas de modelado de objetos 3D se sale fuera del alcance de esta asignatura.

Tanto para el alumno de Ingeniería de Sistemas como de Gestión este capítulo es completamente nuevo. Se recomienda durante el estudio que todas las técnicas descritas a lo largo de este tema, una vez comprendidas, sean probadas sobre diferentes imágenes utilizando para ello el software recomendado.

Objetivos

Conocer algunas de las técnicas utilizadas en la representación de objetos bidimensionales así como aprender a definir el modelo o vector de características que permitirá el reconocimiento del objeto durante la siguiente etapa de procesado.

4.1 Introducción

La imagen binaria obtenida tras el proceso de segmentación generalmente es representada como una matriz, la cual contiene información tanto del objeto como del fondo. Una representación más compacta de la imagen, se puede obtener al almacenar en una estructura de datos, la información de los píxeles que acotan al objeto. De esta forma, bastaría con realizar análisis u operaciones sobre la estructura de datos, para realizar la extracción de los parámetros que definen el objeto, o para aplicar operaciones de modificación de la forma del mismo.

Esto es, se entiende por representación de un objeto al proceso que transforma los píxeles que lo integran en un formato más compacto y de un nivel superior. Así, por ejemplo, una forma eficaz de representar el contorno de un objeto podría ser a través de un polígono.

En este tema se comienzan describiendo algunos de los métodos existentes para representar estructuras bidimensionales y a continuación, se estudiarán algunos de los procedimientos dedicados al cálculo de características discriminantes.

4.2 Representación de estructuras geométricas bidimensionales

Existen dos maneras de representar la forma bidimensional de un objeto: mediante su contorno definido por la curva que limita su forma o mediante su región. A continuación veremos algunas de las técnicas englobadas en ambas estrategias.

4.2.1 Representación del contorno

La representación del contorno de un objeto segmentado en la imagen es el conjunto de píxeles que definen dicho contorno. Esta representación no es eficiente desde el

punto de vista computacional y de almacenamiento de la información al contener demasiada información. Seguidamente, vamos a estudiar varias formas de representar el contorno de un objeto en la imagen que permiten reducir tal información.

Polilíneas

Una curva en el plano puede ser aproximada mediante un conjunto de segmentos de línea los cuales pueden ser definidos mediante sus puntos extremos. Este método consiste en representar la secuencia de segmentos de línea como una lista de puntos. Así, tres puntos x_1, x_2 y x_3 representan una secuencia de segmentos de línea de la forma:

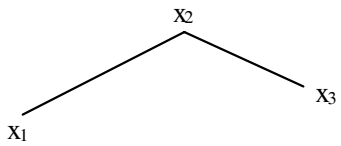


Figura 4.1: Secuencia de segmentos de línea.

Si el primer punto es el mismo que el último se trata de un contorno cerrado.

Las polilíneas pueden aproximar la mayoría de las curvas para cualquier grado de precisión. El problema es encontrar los puntos que producen la mejor polilínea. Encontramos varios métodos para conseguirlo:

- *Técnica de fusión*

  Se trata de ir ajustando los puntos del contorno mediante una recta hasta que el error cometido en el ajuste supere un determinado umbral. Entonces, se almacenan los parámetros de la recta resultante y se repite el proceso con el resto de los puntos pertenecientes al contorno hasta que todos éstos hayan sido tratados. Para calcular los vértices del polígono basta con determinar la intersección de las rectas adyacentes. La figura4.2 ilustra este método.

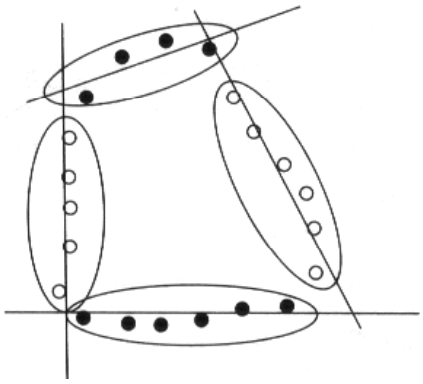


Figura 4.2: Ajuste poligonal de un contorno mediante el método de fusión.

- El principal problema de esta técnica reside en que los vértices calculados no se suelen corresponder con las esquinas del objeto. Esto es debido a que, al llegar a una esquina de la frontera en el proceso de fusionado, sólo tras haber incorporado unos cuantos puntos de la misma no se supera el umbral considerado y, en consecuencia no se comenzará una nueva recta. Esto provocará que las esquinas aparezcan con un cierto desfase dependiente del umbral utilizado durante el proceso.

- *Polilínea de longitud mínima*

  Cada píxel se considera como un cuadrado. El contorno se aproxima a la forma de un hilo elástico colocado en la estructura píxel tal y como se muestra en la figura4.3.

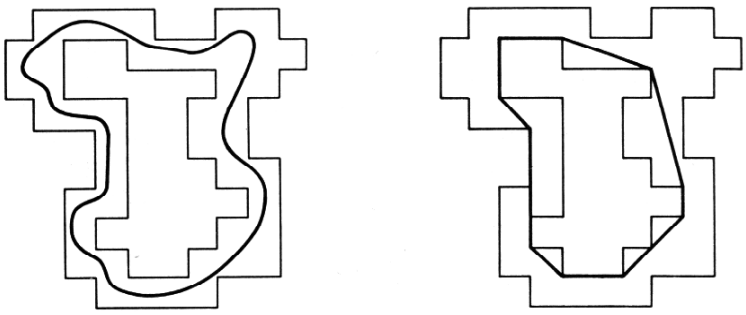


Figura 4.3: Aplicación del método de longitud mínima.

- *Método de división recursiva*

  Consiste en dividir sucesivamente un segmento en dos hasta que se satisfaga un determinado criterio como, por ejemplo, que la distancia de los puntos de un tramo del contorno al segmento que los aproxima no sobrepase un determinado valor. En caso de que lo sobrepase, el punto más alejado es considerado un nuevo vértice por donde se subdivide el segmento en dos. Para el caso de una frontera cerrada, los puntos de comienzo que hay que tomar son los dos puntos de la frontera más lejanos entre sí.

  Este método presenta la ventaja de detectar los puntos de inflexión pero, sin embargo, es muy sensible a puntos falsos y si el contorno tiene concavidades, el proceso se complica.

  El algoritmo que implementa este método viene definido por los siguientes pasos:

  (a) Dado el conjunto de puntos pertenecientes al contorno, dibujar una línea recta entre su punto inicial y final, esto es, entre sus puntos más alejados.

  (b) Para cada uno de estos puntos, calcular la distancia perpendicular a la línea. Si todas ellas son menores a un cierto valor se considera que dicha línea representa a la curva.

  (c) Sino escoger el punto más alejado de la línea y reemplazar esta línea por dos nuevos segmentos de línea (ver figura4.4).

  (d) Aplicar recursivamente el algoritmo a los nuevos segmentos hasta que se consiga el grado de precisión deseado.

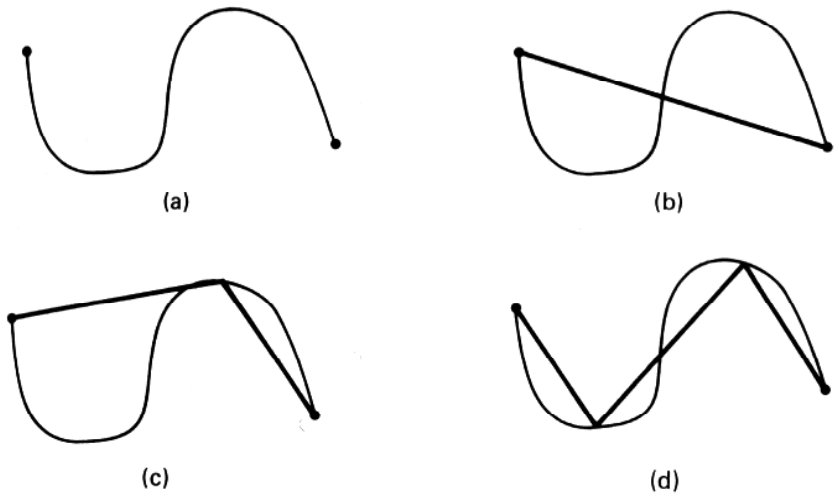


Figura 4.4: Aplicación del método de división recursiva.

Códigos de cadena

Los códigos de cadena se utilizan para representar un contorno como un conjunto de vectores de magnitud y dirección específica, conectados entre sí. Por lo general, esta representación está basada en segmentos de 4 u 8 direcciones, codificadas como se muestra en la figura4.5.

El procedimiento para generar el código de un contorno consiste en elegir un píxel de partida, que pertenezca a dicho contorno, e ir recorriendo en sentido horario la frontera, a la vez que se va anotando el código que mejor se aproxima a ésta. Una manera trivial de hacer esto es unir con un segmento cada par de píxeles adyacentes. Esta solución no es aceptable puesto que el código resultante sería demasiado largo. Un método más aconsejable consiste en tomar una rejilla como la que se muestra en la figura4.6(a) y asignar a cada punto del contorno el nodo de la rejilla que se encuentre más cercano a él como se ilustra en la figura 4.6(b). Seguidamente, el contorno del objeto se puede representar mediante un código de cadena de 4 u 8 direcciones, tal y como se muestra en las figuras4.6(c) y (d), respectivamente.

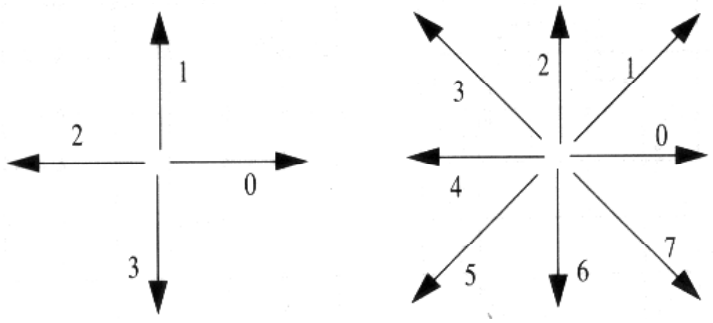
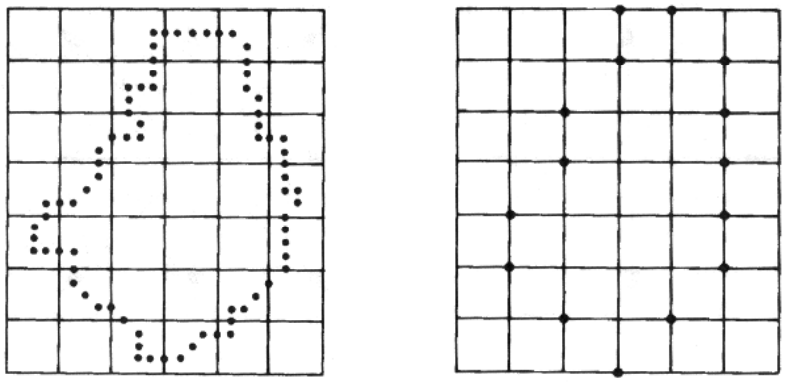
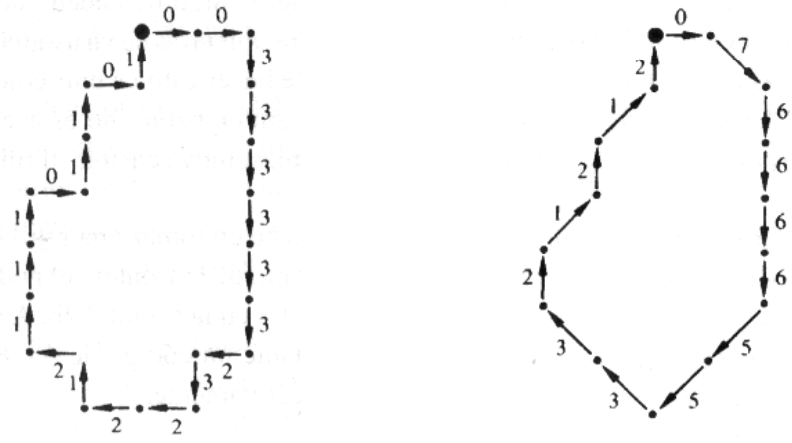


Figura 4.5: Direcciones de un código de cadena de 4 y de 8 direcciones.



(a) Contorno de un objeto (b)Puntos considerados



(c) Código de cadena de 4 direcciones (d)Código de cadena de 8 direcciones

Figura 4.6: Ejemplo de representación utilizando código de cadena

Nótese que dependiendo del punto de comienzo variará el código de cadena resultante. Para evitar este problema se normaliza el código, esto es, una vez obtenido

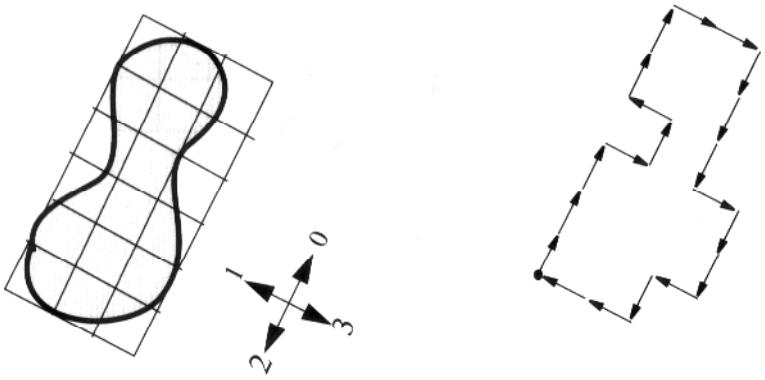
el código es tratado como una secuencia circular y se redefine el punto de comienzo tal que el número entero resultante sea el menor posible.

Por otro lado, para conseguir una representación invariante ante rotación es necesario que el código de cadena se oriente según alguna dirección intrínseca al objeto, por ejemplo, la dirección principal asociada a la máxima dispersión de los píxeles del contorno. De este modo, un método para normalizar la orientación de la rejilla consiste en obtener el rectángulo mínimo que circunscribe al objeto. Las direcciones del código de cadena se tomarán según los ejes de dicho rectángulo tal y como se ilustra en la figura4.7. Para conseguir invarianza respecto al tamaño, el rectángulo se debe dividir siempre en el mismo número de celdas, independientemente del tamaño del objeto.

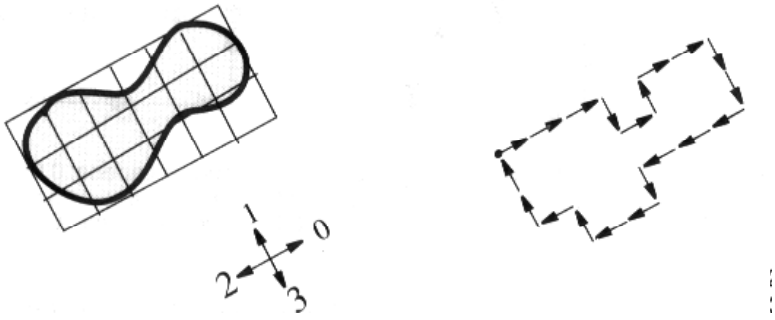
El procedimiento sería el siguiente:

- Obtener el rectángulo mínimo que circunscribe al contorno.
- Determinar el tamaño de la rejilla dependiendo de la precisión que se desea. Cuanto más pequeña sea ésta mayor precisión.
- Obtener el código de cadena.

Una de las ventajas más importantes del código de cadena es que a partir de éste, es inmediato la descripción del objeto mediante características relacionadas con la forma tales como el perímetro, la curvatura, el área, la longitud máxima, etc.



(a) Código de cadena: 000301003322232212 11



(b) Código de cadena: 000301003322232212 11

Figura 4.7: Obtención del código de cadena para un mismo objeto con orientación y tamaño diferentes

Curva $\Psi - s$

La curva $\Psi - s$ es la versión continua de los códigos de cadena. Ψ es el ángulo entre una línea fija y la tangente a la curva que define el contorno de la forma bidimensional en un punto y s es la longitud del arco recorrido. Para un contorno cerrado la función es periódica con un salto discontinuo desde 2π a 0. Líneas rectas horizontales en la curva $\Psi - s$ corresponden con líneas rectas del contorno y líneas rectas no horizontales corresponden con arcos de circunferencia; es decir, cuando Ψ cambia en una proporción constante. Esto es una propiedad muy interesante en la clasificación de polígonos.

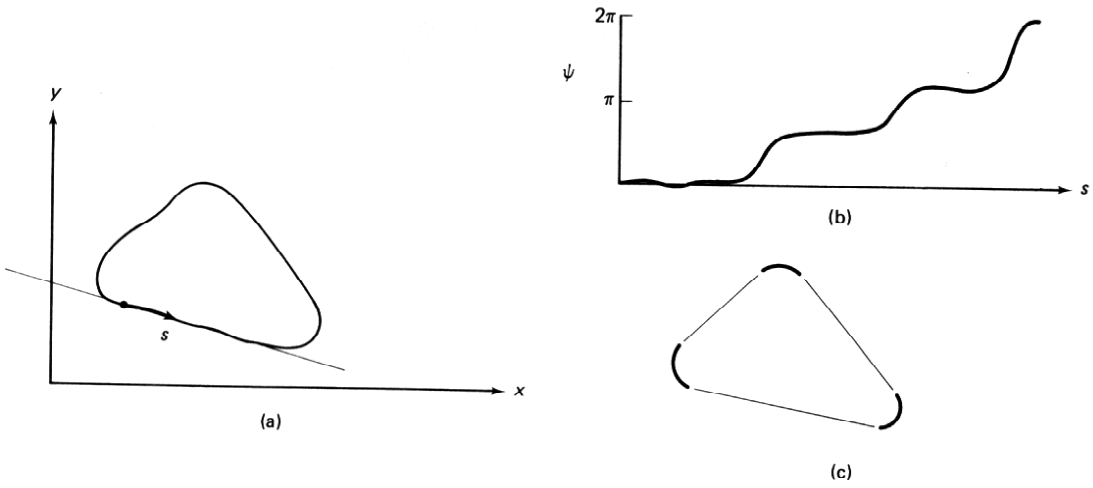


Figura 4.8: (a) Curva con forma triangular. (b) Representación de la curva. (c) Contorno resultante.

Representación polar

La representación polar es una representación unidimensional de un contorno. Existen diferentes formas de generar esta representación. Una de las más simples es la que define la distancia desde el centro de gravedad a un punto del limite en función del ángulo tal y como se muestra en la figura4.9 para el caso de un círculo.

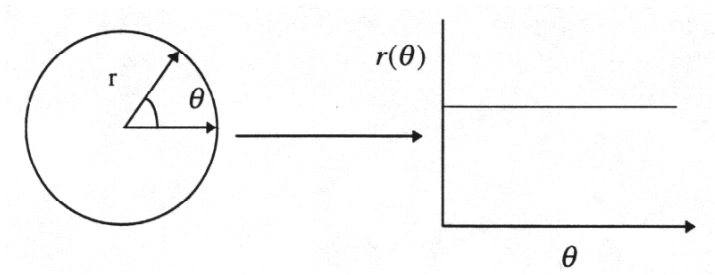


Figura 4.9: Representación polar de un círculo.

La representación polar, aunque invariante a la posición del objeto en la imagen, es dependiente del tamaño y punto de comienzo (punto por donde se empieza a describir el contorno). Sin embargo, es inmediato obtener una representación invariante

a ambos:

- La invarianza al tamaño se consigue dividiendo la función por la distancia máxima al centro de gravedad, de forma que la distancia máxima resultante sea igual a 1.
- La invarianza ante el ángulo de comienzo se logra comenzando la representación por el ángulo para el cual la distancia es máxima.

Curva B-Spline

Otra manera de aproximar el contorno de un objeto es mediante técnicas de interpolación. Un ejemplo son las curvas B-Spline. Estas combinan los efectos de $n+1$ puntos de control (p_i) dados por:

$$\overrightarrow{p}(u) = \sum_{i=0}^{n} \overrightarrow{p}_i N_{i,k}(u) \tag{4.1}$$

El grado de este polinomio está controlado por un parámetro k que es independiente del número de puntos de control. Tenemos que:

$$\begin{aligned} N_{i,1}(u) &= 1 \qquad si\ t_i \leq u \leq t_{i+1} \\ N_{i,1}(u) &= 0 \qquad en\ otro\ caso \end{aligned} \tag{4.2}$$

$$N_{i,k}(u) = \frac{(u-t_i)N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k-1}-t_{i+1}} \qquad con\ \frac{0}{0}=1$$

donde k controla el grado del polinomio resultante en u y así, controla también la continuidad de la curva.

- Para una **curva abierta**, t_i toma los valores:

$$t_i = 0 \qquad si\ i < k$$

$$\begin{aligned} t_i &= i-k+1 \qquad si\ k \le i \le n \\ t_i &= n-k+2 \qquad si\ i > n \qquad con\ 0 \le i \le n+k \end{aligned} \tag{4.3}$$

El rango de la variable paramétrica u es $0 \le u \le n-k+2$.

Cada segmento de curva B-Spline esta influenciada por k puntos de control y por tanto, cada punto de control influye sobre k segmentos de curva.

Las ecuaciones en forma matricial para una B-Spline abierta con $k=3$, son:

$$\overrightarrow{p}_i(u) = \frac{1}{2}\begin{bmatrix} u^2 & u & 1 \end{bmatrix}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \end{bmatrix} = \overrightarrow{U}_3\overrightarrow{M}_3\begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \end{bmatrix} \tag{4.4}$$

para $i \in [1 : n-1]$ y donde:

$$\overrightarrow{U}_3 = \begin{bmatrix} u^2 & u & 1 \end{bmatrix} \text{ y } \overrightarrow{M}_3 = \frac{1}{2}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Para una B-Spline abierta con $k=4$ queda:

$$\begin{aligned} \overrightarrow{p}_i(u) &= \frac{1}{6}\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} p_{i-1} \\ pi \\ p_{i+1} \\ p_{i+2} \end{bmatrix} = \\ &= \overrightarrow{U}_4\overrightarrow{M}_4\begin{bmatrix} p_{i-1} \\ pi \\ p_{i+1} \\ p_{i+2} \end{bmatrix} \end{aligned} \tag{4.5}$$

para $i \in [1 : n-2]$ y donde:

$$\overrightarrow{U}_4 = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \text{ y } \overrightarrow{M}_4 = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

- Para el caso de **curvas cerradas** tenemos para $k=3$:

$$\overrightarrow{p}_i(u) = \overrightarrow{U}_3\overrightarrow{M}_3\begin{bmatrix} \overrightarrow{p}_{(i-1)\bmod(n+1)} \\ \overrightarrow{p}_{i\bmod(n+1)} \\ \overrightarrow{p}_{(i+1)\bmod(n+1)} \end{bmatrix} \qquad para\ i \in [1 : n+1] \tag{4.6}$$

y cuando $k = 4$ tenemos:

$$\overrightarrow{p}_i(u) = \overrightarrow{U}_4 \overrightarrow{M}_4 \begin{bmatrix} \overrightarrow{p}_{(i-1) \bmod(n+1)} \\ \overrightarrow{p}_{i \bmod(n+1)} \\ \overrightarrow{p}_{(i+1) \bmod(n+1)} \\ \overrightarrow{p}_{(i+2) \bmod(n+1)} \end{bmatrix} \qquad para\ i \in [1 : n+1] \qquad (4.7)$$

En la figura4.10 se muestra un ejemplo de una curva B-spline.

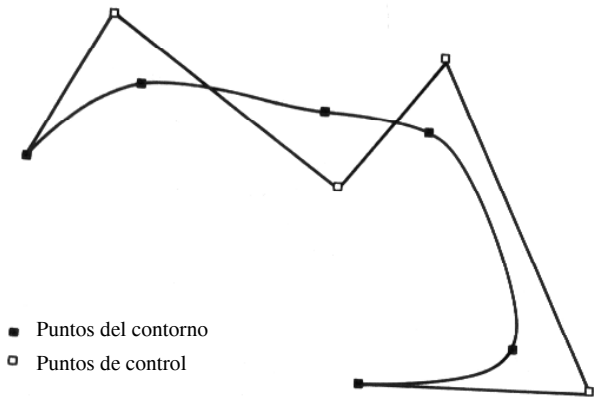


Figura 4.10: Ejemplo de una curva B-spline.

La ventaja del empleo de las curvas B-spline frente a otros tipos de curvas tales como las Spline o las Bezier es que el cambio de uno de sus puntos no afecta a toda la curva sino únicamente a los puntos pertenecientes a su vecindad.

Árbol de rectángulos

Es un árbol binario. En cada nodo del árbol nos encontramos ocho valores diferentes, de los cuales seis definen un rectángulo y dos denotan la dirección de sus hijos (ver figura4.11).

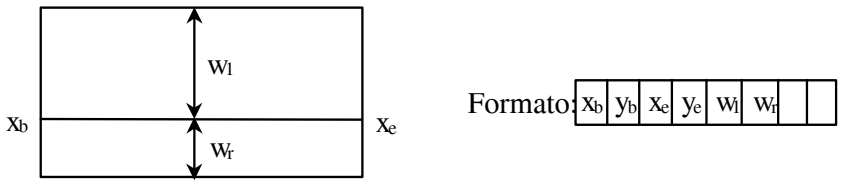


Figura 4.11: Formato de los datos almacenados en un nodo del árbol de rectángulos.

Para la creación del árbol aplicamos el siguiente procedimiento recursivo: se comienza buscando el rectángulo más pequeño con dos lados paralelos al segmento que une el punto x_0 con x_n y que encierra a todos los puntos del contorno. Este rectángulo será el nodo raíz del árbol. Seguidamente se toma el punto x_k que yace sobre uno de los lados del rectángulo y se repite el proceso para los dos segmentos definidos entre los puntos x_0, x_k y x_n. La figura4.12 ilustra gráficamente este método.

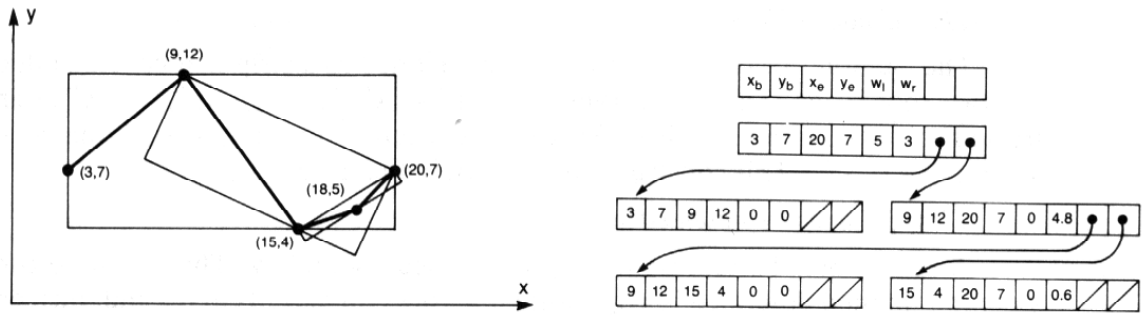


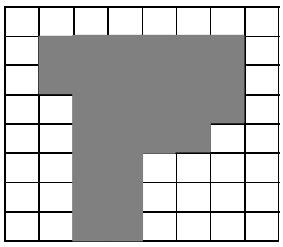
Figura 4.12: Proceso de construcción del árbol.

4.2.2 Representación de regiones

En esta sección se estudiaran algunos de los métodos existentes para la representación de regiones.

Matriz de ocupación espacial

Se trata del método más obvio y simple de representación de regiones y consiste en una matriz cuyos elementos toman el valor de 1 cuando el píxel (x, y) pertenece a la región y 0 en otro caso. En la figura4.13 se muestra un ejemplo de este método.



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

Figura 4.13: Ejemplo de la matriz de ocupación espacial.

Una de las desventajas de este método es que ocupa mucha memoria.

Árboles de cuadrados

En este caso se engloba la región en un cuadrado. Inicialmente, en el nivel más bajo se considera que este cuadrado puede ser: negro, blanco o gris. Es decir, si todos sus píxeles correspondientes son negros o blancos el cuadrado será negro o blanco respectivamente, si algunos de sus píxeles son negros y otros son blancos entonces será gris. Cuando exista uno o mas cuadrados grises se repite el proceso descomponiendo el cuadrado inicial en otros cuatro más pequeños, generando un nivel superior (ver figura4.14). De esta manera se puede decir que construimos una pirámide de niveles.

Para convertir la pirámide en un árbol de cuadrados se realiza el siguiente procedimiento: Partiendo del nivel 0, si un elemento en la pirámide es blanco o negro, éste constituye un elemento terminal del tipo correspondiente en el árbol (negro o blanco). En otro caso, es decir, cuando el elemento es gris, se analiza el siguiente nivel valorándose cada una de sus posibilidades.

En la figura4.15 se representa el árbol de cuadrados resultante para el ejemplo mostrado en la figura4.14.

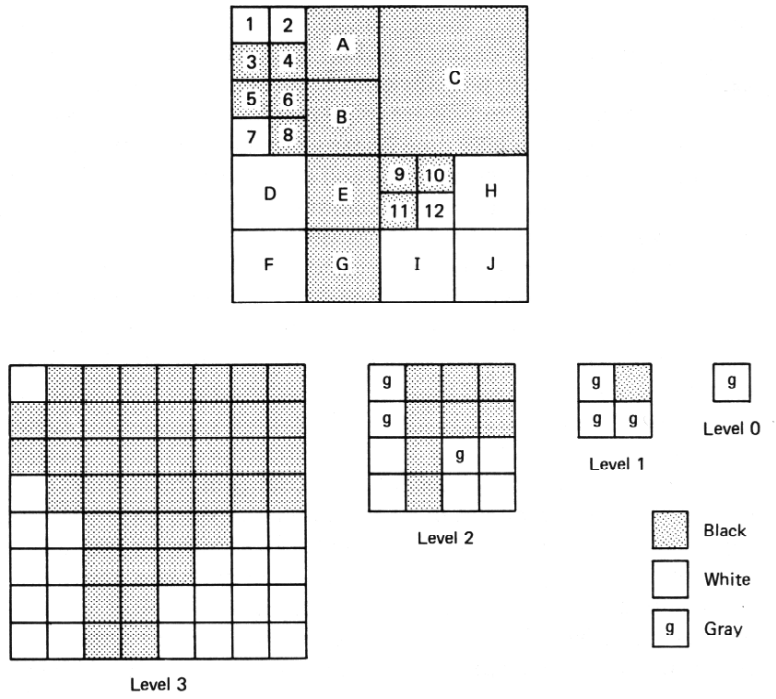


Figura 4.14: Construcción de la pirámide para el método de árboles de cuadrados.

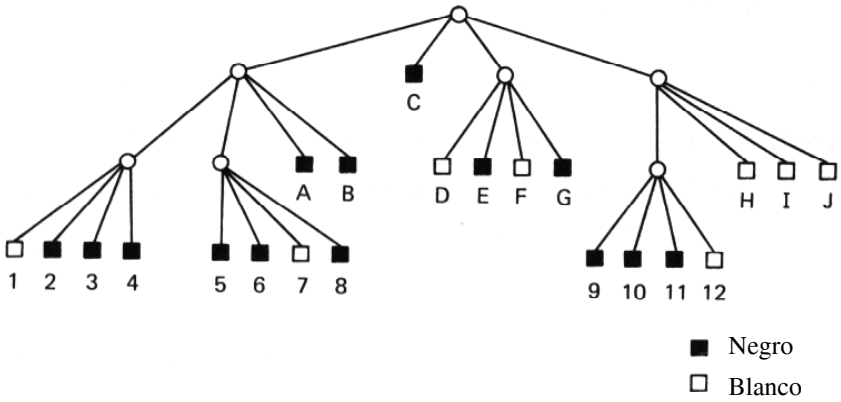


Figura 4.15: Árboles de cuadrados para el ejemplo de la figura.

Técnicas de esqueletizado

La idea básica del esqueleto de una región es la de conservar toda la información sobre la forma y estructura del objeto, eliminando redundancias que no aportan

nada al proceso de reconocimiento. A partir de esta representación debe ser más fácil obtener una descripción invariante al tamaño, posición y orientación.

Un punto pertenece al esqueleto si éste da la mínima distancia para al menos dos puntos del contorno diferentes. En la figura4.16 se muestran varios ejemplos de esta técnica.

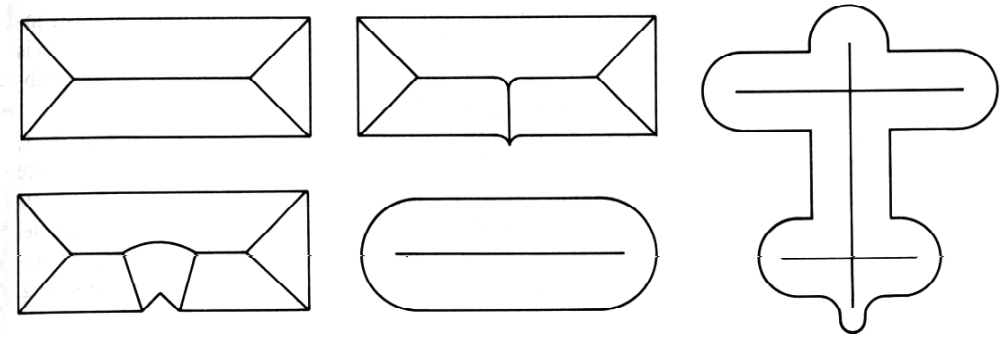


Figura 4.16: Ejemplos del esqueleto de varios objetos.

Esta técnica presenta como principal inconveniente el elevado número de operaciones que implica. Una manera más eficiente de determinar el esqueleto de una región es mediante técnicas de adelgazamiento, si bien el resultado no siempre coincide con la definición dada anteriormente, aunque es igualmente válido.

Los algoritmos de adelgazamiento consisten en ir borrando iterativamente píxeles del exterior sin llegar a eliminar aquellos terminales, ni que se produzca la desconexión de la región. Estos algoritmos suelen estar basados en técnicas morfológicas de erosión y dilatación o en procedimientos específicos.

4.3 Cálculo de características discriminantes

Existen varios procedimientos para calcular características discriminantes de los contornos cerrados. Algunos de ellos se inspiran en el concepto de momento de una función bidimensional $f(x, y)$ acotada y cerrada en el plano x, y. Otros utilizan las

propiedades topológicas de los objetos. Un enfoque diferente aparece cuando se emplean las componentes frecuenciales del contorno del objeto calculadas a partir de la transformada de Fourier.

4.3.1 Características discriminantes basadas en los momentos

Dada una función $f(x, y)$ acotada, se define el momento general de orden p, q como la integral doble:

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{4.8}$$

Para una función acotada en el plano existen infinitos momentos generales obtenidos haciendo p y q de cero a infinito.

El interés de estos momentos generales desde el punto de vista de la caracterización discriminante de los contornos de los objetos es que, tal como veremos a continuación, estos contornos pueden modelarse como un tipo especial de funciones $f(x, y)$ acotadas y, por ende, se pueden calcular sus momentos generales. En este sentido, dada una función acotada $f(x, y)$ existe un conjunto único de momentos y viceversa. Es decir, dado un conjunto de momentos generales se puede reconstruir una función $f(x, y)$ única. Simbólicamente:

$$f(x, y) \leftrightarrow \{m_{p,q}\} \quad p, q = 0, 1, 2...\infty$$

Obviamente, esta correspondencia biunívoca no presentaría ningún interés práctico, a menos que pudiera reducir el número de momentos generales a una cantidad manejable.

El problema de la reconstrucción de una función $f(x, y)$ a partir de un conjunto finito de sus momentos generales es obligado planteárselo para cada caso específico.

Para aplicar esta idea al reconocimiento automático de los contornos de los objetos en una imagen digital es preciso pasar de una integral doble a un doble sumatorio

puesto que la función $f(x,y)$ es, ahora, una función $I(x,y)$ acotada que toma valores distintos de cero únicamente en el contorno del objeto y en su interior. Así, los momentos generales discretos de la función $I_0(x,y)$ serán:

$$m_{p,q} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} x^p y^q I_0(x,y) \qquad con\ p = 0,1,...,\infty \ \ y \ \ q = 0,1,....,\infty \tag{4.9}$$

Dependiendo de cómo se defina $I_0(x,y)$, se tendrán cuatro posibles momentos generales:

1. $I_0(x,y) = 1$, *contorno*

   $I_0(x,y) = 0$, *resto*

2. $I_0(x,y) = I(x,y)$, *contorno*

   $I_0(x,y) = 0$, *resto*

3. $I_0(x,y) = 1$, *objeto*

   $I_0(x,y) = 0$, *resto*

4. $I_0(x,y) = I(x,y)$, *objeto*

   $I_0(x,y) = 0$, *resto*

Es decir, se puede definir una función acotada $I_0(x,y)$ que tome los valores $(1,0)$ o bien los valores $(I(x,y),0)$. Siendo $I(x,y)$ el nivel de intensidad luminosa en la escala de grises. Igualmente se puede establecer el campo de existencia de la función $I_0(x,y)$ bien como el contorno del objeto o bien como el contorno más su interior.

Los momentos generales que se obtienen a partir del contorno son muy sensibles al ruido y a pequeñas variaciones en la forma del contorno. Por el contrario, los momentos generales basados en el contorno más su interior presentan una mayor robustez. Por otro lado, para caracterizar la forma de un objeto no es necesario manejar funciones $I_0(x,y)$ multievaluadas, siendo suficiente que tomen un valor único,

digamos la unidad. En consecuencia, para la obtención de características discriminantes de un contorno cerrado es habitual trabajar con la opción 3.

Un momento de gran interés es el momento de orden cero (el orden de un momento viene dado por la suma de los índices p y q):

$$m_{00} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I_0(x,y) \tag{4.10}$$

que claramente coincide con el área del objeto. El área física se puede calcular sin más que multiplicar m_{00} por el área física de un píxel, lo cual obliga al calibrado previo de la cámara. Afortunadamente, desde el punto de vista del reconocimiento de objetos veremos que no es necesario pasar a las dimensiones físicas de los objetos, ya que es posible obtener momentos invariantes a traslaciones, giros y tamaños relativos de los objetos.

Los momentos de orden 1, m_{01} y m_{10}, junto con m_{00}, determinan el llamado centro de gravedad del objeto:

$$\overline{x} = \frac{m_{10}}{m_{00}} = \frac{\sum\sum xI(x,y)}{\sum\sum I(x,y)} \tag{4.11}$$

$$\overline{y} = \frac{m_{01}}{m_{00}} = \frac{\sum\sum yI(x,y)}{\sum\sum I(x,y)} \tag{4.12}$$

El inconveniente de los momentos generales radica en el hecho de que varían con la posición del objeto dentro del plano de la imagen, así como con el tamaño relativo del objeto, que depende de la distancia entre la cámara y el objeto. Los cambios en la posición del objeto se deben exclusivamente a las operaciones de giro y traslación. En cuanto a los cambios de tamaños se puede modelar como una operación de homotecia.

Por tanto, vamos a transformar sucesivamente los momentos generales para hacerlos invariantes a traslaciones, rotaciones y a homotecias.

Momentos invariantes a traslaciones

Los momentos generales se pueden hacer invariantes a traslaciones en el plano sin más que referirlos al centro de gravedad (x_c, y_c), obteniéndose los llamados momentos centrales:

$$\mu_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (x - \overline{x})^p (y - \overline{y})^q I_0(x, y) \tag{4.13}$$

Puesto que el centro de masas ocupa siempre la misma posición relativa respecto al resto de los puntos del objeto, los momentos centrales no varían ante traslaciones de los objetos.

El cálculo directo de los momentos centrales , en principio, es complejo, según se desprende de la expresión (4.13). Afortunadamente, es posible obtener una formula general válida para su cálculo directo en función de los momentos generales:

$$\mu_{pq} = \sum_{r=0}^{p} \sum_{s=0}^{q} \begin{pmatrix} p \\ r \end{pmatrix} \begin{pmatrix} q \\ s \end{pmatrix} (-\overline{x})^r (-\overline{y})^s m_{p-r,q-s} \tag{4.14}$$

De este modo, aplicando esta expresión se obtendrían los siguientes momentos:

$$\mu_{00} = m_{00};\ \mu_{01} = \mu_{10} = 0 \tag{4.15}$$

$$\mu_{11} = m_{11} - \frac{m_{10} m_{01}}{m_{00}} \tag{4.16}$$

$$\mu_{20} = m_{20} - \frac{m_{10}^2}{m_{00}} \tag{4.17}$$

$$\mu_{02} = m_{02} - \frac{m_{01}^2}{m_{00}} \tag{4.18}$$

Momentos invariantes ante rotaciones

La idea básica para lograr momentos invariantes a los giros que se puedan aplicar a una objeto bidimensional es definir una dirección única de referencia del objeto. Esta dirección de referencia va a ser el denominado eje de mínima inercia, que siempre será el mismo para un objeto bidimensional específico. Observando la figura4.17, el problema se limita a calcular el ángulo θ que forma el eje de mínima inercia con uno de los ejes coordenados de la imagen.

Veamos como calcular el ángulo θ:

Sabemos que la ecuación de la recta que pasa por un punto (x_1, y_1) formando un ángulo θ con el eje Y viene dada por:

$$(y - y_1)\cos\theta = (x - x_1)\text{sen}\theta \tag{4.19}$$

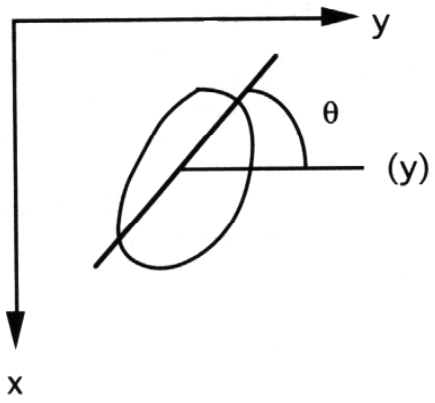


Figura 4.17: Eje de mínima inercia.

El momento de inercia de un sólido bidimensional respecto a la recta anterior es:

$$I = \sum_x \sum_y \left[(x - x_1)\text{sen}\theta - (y - y_1)\cos\theta\right]^2 f(x, y) \tag{4.20}$$

siendo $f(x, y)$ la función definida por el sólido. Obviamente, particularizando esta definición al caso que nos ocupa, la función $f(x, y)$ será aquí la función binaria $I_0(x, y)$ definida por el objeto.

Para minimizar el momento de inercia, se deberá resolver el sistema de dos ecuaciones:

$$\frac{\partial I}{\partial x} = 0; \qquad \frac{\partial I}{\partial y} = 0 \tag{4.21}$$

Desarrollando se llegaría a:

$$m_{10}\text{sen}\theta - x_1 m_{00}\text{sen}\theta - m_{01}\cos\theta + y_1 m_{00}\cos\theta = 0 \tag{4.22}$$

que en virtud de la definición del centro de gravedad se puede reescribir como:

$$(\overline{x} - x_1)\text{sen}\theta - (\overline{y} - y_1)\cos\theta = 0 \tag{4.23}$$

De lo cual se deduce que la recta de mínima inercia pasa por el centro de gravedad $(\overline{x}, \overline{y})$ del objeto, ya que cumple la ecuación de la recta dada por la expresión (4.19):

$$\overline{x} = x_1; \qquad \overline{y} = y_1 \tag{4.24}$$

Para obtener el ángulo θ se aplicaría la condición del mínimo:

$$\frac{\partial I}{\partial \theta}(\overline{x}, \overline{y}) = 0 \tag{4.25}$$

obteniéndose la ecuación:

$$\tan^2\theta + (\mu_{20} - \mu_{02})\frac{\tan\theta}{\mu_{11}} + 1 = 0 \tag{4.26}$$

dando lugar al ángulo θ tal que:

$$\tan 2\theta = \frac{2\mu_{11}}{(\mu_{20} - \mu_{02})} \tag{4.27}$$

Una vez obtenida la dirección de referencia, en principio habría que girar el objeto un ángulo θ y calcular a continuación los momentos invariantes a giros. Pero este proceso es muy complejo y haría que los tiempos de cálculo fueran elevados.

No obstante, es posible expresar los momentos invariantes a giros ϕ_{pq} en función de los momentos centrales:

$$\phi_{pq} = \sum_{r=0}^{p}\sum_{s=0}^{q} (-1)^{q-s} \begin{pmatrix} p \\ r \end{pmatrix} \begin{pmatrix} q \\ s \end{pmatrix} (\cos\theta)^{p-r+s} (\text{sen}\theta)^{q-s+r} \mu_{p-r+q-s,r+s} \tag{4.28}$$

que aunque aparentemente engorrosa es de cálculo inmediato.

Momentos invariantes a homotecias

Podemos conseguir momentos invariantes a homotecias, es decir, a cambios en el tamaño de los objetos mediante la expresión:

$$M_{pq} = \frac{\phi_{pq}}{m_{00}^{\gamma}}; \quad con \;\; \gamma = \frac{p+q}{2} \;\; y \;\; p, q = 0, 1, ...\infty \tag{4.29}$$

El problema práctico fundamental es determinar el número de momentos invariantes que permitan caracterizar a cada uno de los objetos del universo de trabajo. Es decir, habría que evaluar el grado de reconstrucción que se obtendría con un conjunto finito de momentos. Por supuesto, este proceso de reconstrucción dependerá de la forma de los objetos de manera que cuanto más irregular sean sus contornos serán necesarios mayor número de momentos. Pero, más importante que el problema de reconstrucción es el de discriminación que normalmente es menos exigente que la reconstrucción, en particular cuando los objetos que intervienen difieren apreciablemente.

Resumiendo, dado un objeto en forma de una función bidimensional $I_0(x, y)$ que toma el valor unidad en su contorno y en el interior del mismo, y el valor nulo fuera de su región, se obtendría un conjunto finito de características invariantes a traslaciones, rotaciones y homotecias siguiendo el siguiente proceso secuencial:

$$m_{pq} \xrightarrow{(\overline{x},\overline{y})} \mu_{pq} \xrightarrow{\theta} \phi_{pq} \xrightarrow{\frac{1}{m_{00}^{\gamma}}} M_{pq}$$

siendo un problema práctico a resolver en cada aplicación específica el número de estos momentos invariantes a considerar como elementos finales del vector de características. Se ha de comentar que se comienza siempre determinando los momentos de menor orden, siguiendo con los de órdenes superiores ya que es en los primeros donde están concentradas las principales características de la forma de un contorno cerrado.

Cálculo de los momentos generales a partir del código de cadena

Una vez planteado y desarrollado el fundamento teórico de la aplicación de los momentos de una función bidimensional acotada $I_0(x, y)$, vamos a ver cómo realizar su cálculo a partir, precisamente, del código cadena del contorno de $I_0(x, y)$.

En el siguiente desarrollo se utiliza un conocido resultado de análisis matemático:

Sean dos funciones bidimensionales P(x,y) y Q(x,y) continuas y acotadas en

una región A del plano, estando A limitada por la curva C, se cumple la siguiente igualdad:

$$\iint_A \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right) dxdy = \int_C Pdx + Qdy \tag{4.30}$$

Este teorema permite pasar la integral doble extendida sobre la región A a una integral curvilínea sobre una curva cerrada C. Este resultado nos va a permitir calcular los momentos generales como integrales curvilíneas extendidas sobre el contorno de un objeto y más concretamente, utilizando el código de cadena que es una aproximación poligonal del contorno.

Si en la expresión (4.30) hacemos:

$$P = -x^p y^{q+1}; \quad Q = x^{p+1} y^q \tag{4.31}$$

entonces

$$\frac{\partial P}{\partial y} = -(q+1)x^P y^Q; \quad \frac{\partial Q}{\partial x} = (p+1)x^p y^q \tag{4.32}$$

y por tanto

$$\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = (p+q+2)x^P y^Q \tag{4.33}$$

Sustituyendo este resultado en la ecuación (4.30) y recordando la definición de momento general de orden p+q tenemos:

$$m_{pq} = \iint_A x^P y^Q dxdy = \frac{1}{p+q+2} \int_C -x^P y^{q+1} dx + x^{P+1} y^q dy \tag{4.34}$$

en donde la función $I_0(x, y)$ asociada al objeto vale la unidad en el propio objeto y cero en el resto del plano.

La integral curvilínea dada por la expresión (4.34) puede calcularse en función del código de cadena del contorno dentro de una imagen digital. Así, la ecuación de una recta genérica que une dos píxeles consecutivos del contorno es:

$$y - y_c = \frac{\Delta y_c}{\Delta x_c}(x - x_c) = m(x - x_c) \tag{4.35}$$

siendo

$$\Delta y_c = y_c - y_{c-1}; \quad \Delta x_c = x_c - x_{c-1} \tag{4.36}$$

Haciendo tender x a x_c e y a y_c se puede escribir $dy = mdx$, que junto con la coordenada y despejada de la ecuación (4.35), se tiene:

$$y = y_c + m(x - x_c) \tag{4.37}$$

lo cual permite expresar la integral curvilínea que calcula el momento general en función de x exclusivamente.

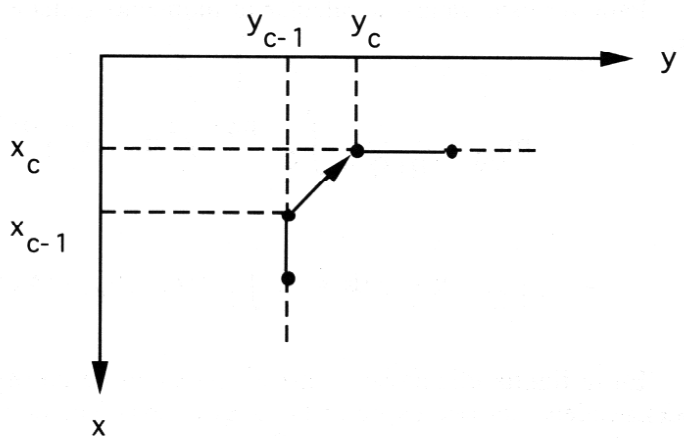


Figura 4.18: Tramo del contorno de un objeto dentro de una imagen digital.

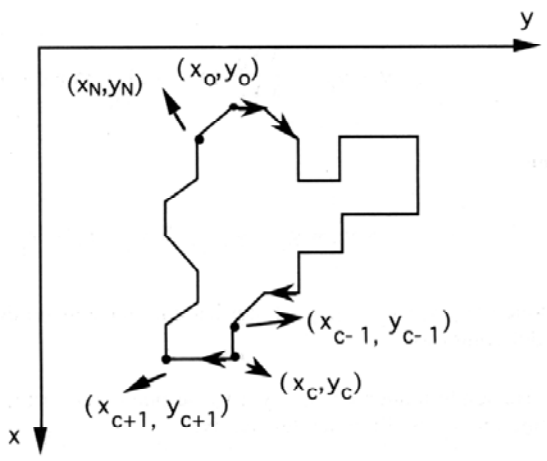


Figura 4.19: Contorno de un objeto definido mediante código de cadena.

Para entender esto, vamos a calcular el momento general de orden 0 m_{00}:

$$m_{00} = \frac{1}{p+q+2}\int_C x^{P+1}y^q dy - x^P y^{q+1}dx = \frac{1}{2}\int_C xdy - ydx = \tag{4.38}$$

$$\frac{1}{2}\int_C mx - [y_c + m(x - x_c)]dx$$

Cuando se emplean códigos de cadena es posible expresar esta integral curvilínea como un sumatorio. Por tanto, podemos escribir:

$$m_{00} = \frac{1}{2}\sum_{C=1}^{N}\int_{x_{c-1}}^{x_c}(mx - [y_c + m(x - x_c)]dx = \frac{1}{2}\sum_{C=1}^{N}\int_{x_{c-1}}^{x_c}(mx_c - y_c)dx \quad (4.39)$$

Obsérvese que el índice del sumatorio se refiere al orden del enlace del código cadena. Desarrollando esta expresión se obtiene:

$$\begin{aligned} m_{00} &= \frac{1}{2}\sum_{C=1}^{N}\int_{x_{c-1}}^{x_c}(mx_c - y_c)dx = \frac{1}{2}\sum_{C=1}^{N}(mx_c - y_c)(x_c - x_{c-1}) = \quad (4.40)\\ &= \frac{1}{2}\sum_{C=1}^{N}\left(\frac{\Delta y_c}{\Delta x_c}x_c - y_c\right)\Delta x_c = \frac{1}{2}\sum_{C=1}^{N}(x_c\Delta y_c - y_c\Delta x_c) \end{aligned}$$

Nótese como dado el código de cadena del contorno de un objeto es inmediato obtener el sumatorio anterior:

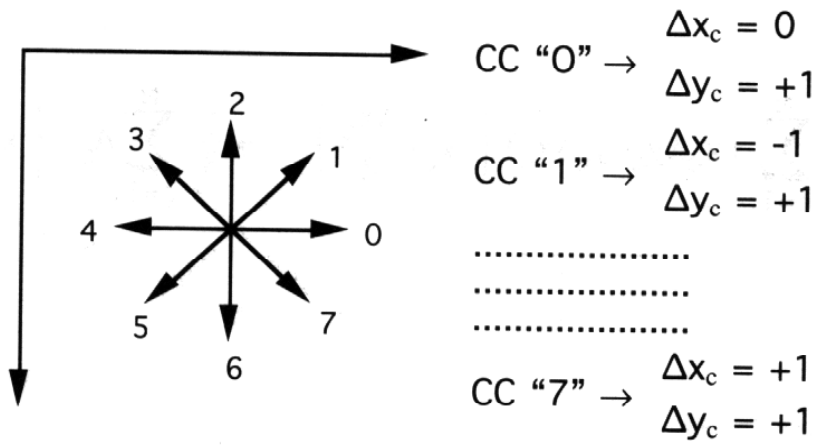


Figura 4.20: Obtención de los incrementos.

Veamos ahora como calcular los momentos de orden uno:

$$m_{10} = \iint_A x\ dx\ dy = \frac{1}{2+1+0}\int_C x^2dy - yxdx \quad (4.41)$$

Aplicando los cambios $dy = mdx$; $y = y_c + m(x - x_c)$, se obtiene:

$$\begin{aligned} m_{10} &= \frac{1}{3}\sum_{C=1}^{N}\int_{x_{c-1}}^{x_c}[mx^2 - xy_c - mx(x - x_c)]dx = \\ &= \frac{1}{3}\sum_{C=1}^{N}\int_{x_{c-1}}^{x_c} x(mx_c - y_c)dx = \frac{1}{3}\sum_{C=1}^{N}(mx_c - y_c)\frac{x^2}{2}|_{x_{c-1}}^{x_c} = \\ &= \frac{1}{3}\sum_{C=1}^{N}(x_c\Delta y_c - y_c\Delta x_c)\left(x_c - \frac{\Delta x_c}{2}\right) \end{aligned} \tag{4.42}$$

Análogamente:

$$m_{01} = \frac{1}{3}\sum_{C=1}^{N}(x_c\Delta y_c - y_c\Delta x_c)\left(y_c - \frac{\Delta y_c}{2}\right) \tag{4.43}$$

Operando de manera análoga se obtiene para los momentos de orden superior las siguientes expresiones:

$$m_{20} = \frac{1}{4}\sum_{C=1}^{N}(x_c\Delta y_c - y_c\Delta x_c)\left(x_c^2 - x_c\Delta x_c + \frac{1}{3}\Delta x_c^2\right) \tag{4.44}$$

$$m_{11} = \frac{1}{4}\sum_{C=1}^{N}(x_c\Delta y_c - y_c\Delta x_c)\left[x_c y_c - \frac{1}{2}(x_c\Delta y_c + y_c\Delta x_c) + \frac{1}{3}\Delta x_c\Delta y_c\right] \tag{4.45}$$

$$m_{02} = \frac{1}{4}\sum_{C=1}^{N}(x_c\Delta y_c - y_c\Delta x_c)\left(y_c^2 - y_c\Delta y_c + \frac{1}{3}\Delta y_c^2\right) \tag{4.46}$$

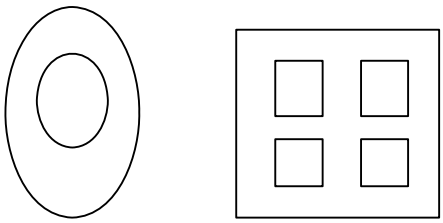
4.3.2 Descriptores topológicos

Las propiedades topológicas son usadas para la descripción global de regiones en la imagen, esto es, propiedades de una figura que no se ven afectadas por las deformaciones. Por ejemplo, una propiedad topológica utilizada es el número de componentes conectadas. Dada una región en una imagen, una componente conectada de dicha región es una subregión de tamaño máximo en la cual todos los puntos se pueden interconectar mediante una curva incluida en dicha subregión.

Pues bien, para un conjunto de regiones conectadas resulta útil emplear el **número de Euler E** como descriptor. Éste se define como la diferencia entre el número de regiones conectadas C y el número de huecos existentes H.

$$E = C - H \tag{4.47}$$

Así, por ejemplo la figura4.21(a) tiene un número de Euler igual a 0 ya que el número de componentes conectadas es 1 y el número de huecos es también 1, y para la figura4.21(b) es igual a -3 ya que C = 1 y H = 4.



(a) E=C-H=0 (b)E=C-H=-3

Figura 4.21: Ejemplos de aplicación del número de Euler

4.3.3 Descriptores de Fourier

La transformada discreta de Fourier permite extraer las componentes en frecuencia de una curva discreta periódica cualquiera. Puesto que el contorno de un objeto es una curva cerrada, y por tanto periódica, ésta puede muestrearse con un periodo determinado y, transformar dichos puntos mediante la transformada de Fourier discreta al dominio de frecuencias donde se representan la amplitud o energía de cada componente frecuencial. Se denomina descriptor de Fourier al conjunto de puntos en el espacio de frecuencias, resultantes de transformar mediante la transformada de Fourier discreta dichos puntos del contorno. La transformada inversa de Fourier discreta proporciona los puntos originales del contorno.

Software

Ver capítulo 2.

Conocimientos nuevos adquiridos

El alumno a lo largo del tema ha estudiado algunas de las técnicas que se aplican para la representación de objetos bidimensionales y definición de modelos útiles para el reconocimiento. No sólo es necesario que haya aprendido cada una de estas técnicas sino que también debe saber cuando resultará más conveniente su empleo.

Bibliografía complementaria

- Capítulo 6 de M. Sonka, V. Hlavac y R. Boyle: "Imagen Processing, Analysis and Machine Vision". Chapman & Hall Computing, 1993. [10].
- Capítulos 8 y 9 de D. H. Ballard y C. M. Brown: " Computer Vision". Prentice-Hall, 1982. [1].
- Capítulo 7 de J. González: "Visión por computador". Paraninfo, 1999. [5].
- Capítulo 13 de D. Maravall: "Reconocimiento de formas y visión artificial". Rama 1993. [7]
- Capítulo 8 de A. De la Escalera: "Visión por computador. Fundamentos y métodos". Prentice Hall, 2001. [4].
- Capítulo 8 de R. C. González y P. Wintz: "Digital Image Processing". Addison-Wesley Publishing Company, 1987. [6].

Actividades

- Realizar y comprender los ejercicios resueltos suministrados por el equipo docente.
- Desarrollar utilizando cualquiera de las herramientas software los distintos algoritmos estudiados a lo largo de la asignatura. Ello permitirá afianzar conocimientos y detectar la dificultad del proceso en algunos casos. Para ello se suministrará al alumno, junto con el software, una biblioteca de imágenes con las que podrá trabajar.

Autoevaluación

Dada una imagen, el alumno debe de saber qué método aplicar para representar y modelar los objetos presentes en la imagen.

También puede realizar las cuestiones y problemas de exámenes de los cursos anteriores.

Problemas resueltos

PROBLEMA 1

Sea la imagen digital mostrada en la figura adjunta. Se pide:

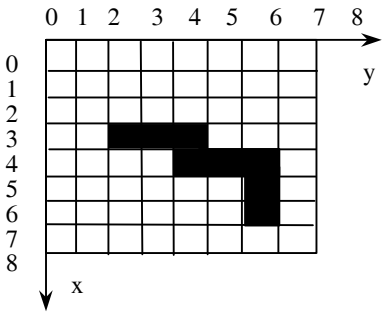


Figura 4.22: Ejemplo de un objeto en una imagen digital.

1. Definir el código de cadena del contorno del objeto utilizando 4 direcciones.
2. Empleando el código anterior determinar una característica del objeto que sea invariante a traslaciones y otra característica que sea invariante a rotaciones.

¿**Qué se pretende con este problema?** Este problema muestra cómo se determina el código de cadena del contorno de un objeto y a continuación cómo se calculan algunas características discriminantes invariantes a traslaciones y rotaciones. Con esto se pretende mostrar un ejemplo práctico de lo aprendido en la parte teórica.

Solución

- *Código de cadena*

Aplicando la definición de los códigos de cadena se tendría la siguiente situación:

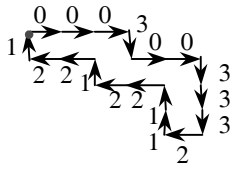


Figura 4.23: Código de cadena de 4 direcciones.

- El código de cadena resultante cuando se toma como punto de comienzo el marcado en la figura4.23 y siguiendo el sentido horario es:

  000300333211221221

  Si se hubiera elegido otro punto de comienzo variaría el código de cadena anterior. La forma de evitar este problema, tal y como se indicó en la parte teórica, consiste en seleccionar el punto de comienzo de forma que el código de cadena resultante sea el menor entero posible. En este caso coincide, precisamente, con el código de cadena determinado anteriormente.

- *Características invariantes a traslaciones*

  Los momentos generales referidos al centro de gravedad del objeto (momentos generales) son características invariantes a traslaciones. Para calcularlos utilizaremos las expresiones (4.15), (4.16), (4.17) y (4.18).

  Construimos la siguiente tabla relativa a los incrementos:

Enlaces	$x_c\Delta y_c$	$y_c\Delta x_c$	$x_c\Delta y_c - y_c\Delta x_c$
000	$3 \cdot 3$	$5 \cdot 0$	9
3	$4 \cdot 0$	$5 \cdot 1$	-5
00	$4 \cdot 2$	$7 \cdot 0$	8
333	$7 \cdot 0$	$7 \cdot 3$	-21
2	$7 \cdot (-1)$	$6 \cdot 0$	-7
11	$5 \cdot 0$	$6 \cdot (-2)$	12
22	$5 \cdot (-2)$	$4 \cdot 0$	-10
1	$4 \cdot 0$	$4 \cdot (-1)$	4
22	$4 \cdot (-2)$	$2 \cdot 0$	-8
1	$3 \cdot 0$	$2 \cdot (-1)$	2
$\sum$			-16

Tabla 4.1: Incrementos resultantes

Aplicando las ecuaciones obtenidas en la parte teórica para el cálculo de los momentos centrales de orden cero se obtiene:

$$\begin{aligned} \mu_{00} &= m_{00} = \frac{1}{2}\sum_{C=1}^{N} |x_c\Delta y_c - y_c\Delta x_c| = \frac{16}{2} = 8 \\ \mu_{01} &= \mu_{10} = 0 \end{aligned}$$

Dejamos como ejercicio numérico para el lector el cálculo de los momentos generales de orden uno.

- *Características invariantes a rotaciones*

  Aplicando la ecuación (4.28) para el cálculo de los momentos invariantes a las rotaciones se tiene:

  $$\phi_{00} = \mu_{00} = 8$$

  Se recomienda al lector el cálculo de los momentos de orden superior siendo necesario para ello determinar el ángulo θ aplicando la ecuación (4.27).

PROBLEMA 2

Determinar la apariencia de la representación polar del cuadrado mostrado en la figura adjunta.

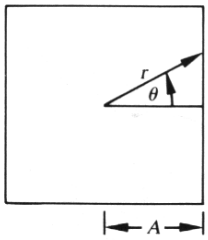


Figura 4.24:

¿**Qué se pretende con este problema?** Este problema pretende mostrar al alumno como la representación polar no solo se trata de una representación de formas bidimensionales sino que también nos permite reconocer objetos.

Solución

Si suponemos la representación polar definida por la distancia desde el punto de gravedad a un punto del limite en función del ángulo θ, se tiene:

$$r(\theta) = A \ \sec\theta$$

Su representación gráfica es de la forma:

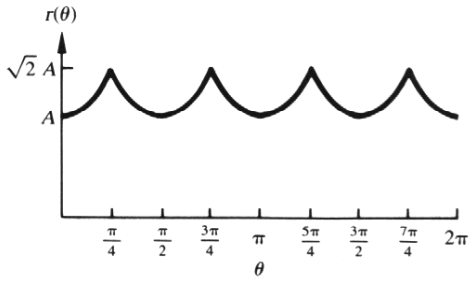


Figura 4.25:

PROBLEMA 3

Sea la imagen digital mostrada en la figura adjunta. Se pide representar el objeto existente utilizando los tres métodos estudiados para la representación de regiones.

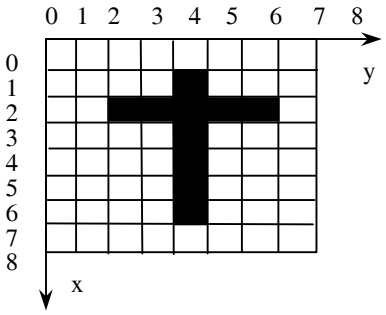


Figura 4.26:

¿**Qué se pretende con este problema?** Este problema pretende mostrar al alumno la aplicación de las técnicas descritas en la parte teórica para la representación de regiones.

Solución

- *Matriz de ocupación espacial*

  La matriz resultante de asignar al punto (x, y) el valor de 1 si pertenece a la región y 0 en otro caso es:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- *Árboles de cuadrados*

  Inicialmente se considera la imagen de partida englobada en un cuadrado. Si todos sus píxeles son negros, blancos o grises el proceso termina. En caso contrario se descompone en cuatro cuadrados repitiéndose el proceso.

  De este modo, la descomposición de la imagen de partida en cuadrados se muestra en la figura4.27 y el árbol de cuadrados resultante en la figura4.28.

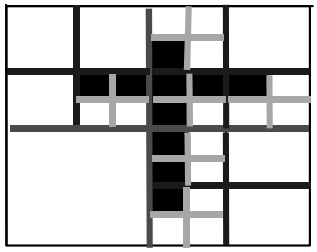


Figura 4.27: Descomposición en cuadrados.

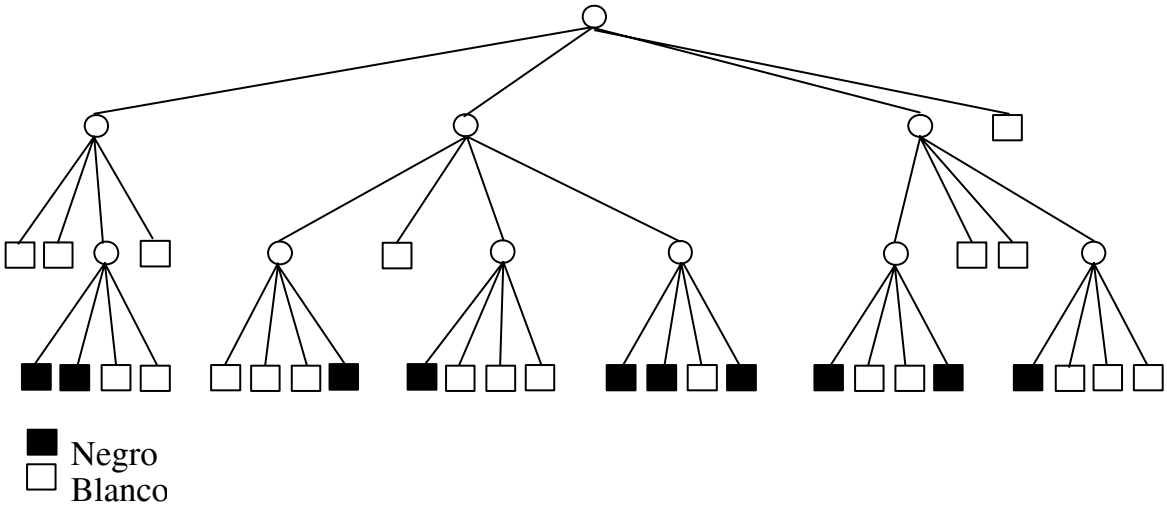


Figura 4.28: Árbol de cuadrados resultante.

- *Técnicas de esqueletizado*

  En la figura4.29 se muestra el esqueleto del objeto detectado en la figura4.29.

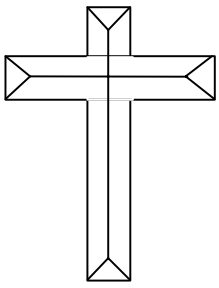


Figura 4.29: Esqueleto resultante.

Capítulo 5

Reconocimiento de objetos e Interpretación

5.1 Introducción al reconocimiento de objetos

El reconocimiento es el proceso gracias al cual asociamos a cada objeto o región de la imagen, un significado, esto es, un objeto del mundo real. Dentro de los sistemas dedicados al reconocimiento encontramos diversas técnicas: reconocimiento de patrones, correspondencia de grafos, redes neuronales, algoritmos genéticos, simulated annealing. Este capítulo se centra en algunos de los métodos utilizados en el reconocimiento de patrones mientras que las redes neuronales serán estudiadas en el capítulo 6. Información relativa a los otros métodos puede ser hallada en diferentes fuentes bibliográficas tales como [1], [10].

A través de las técnicas dedicadas a la extracción de características, descritas en el capítulo anterior, se definen distintas posibilidades de representación y caracterización de las regiones presentes en la imagen de forma que los objetos quedan representados mediante modelos definidos bien mediante un vector de características o bien mediante algún tipo de representación estructural. La primera tarea que debe resolverse en el problema de reconocimiento de patrones es la selección de un vector de características que permita realizar el reconocimiento a partir de los modelos de los posibles objetos. Cuanto mayor sea el vector de características empleado mejor

quedarán definidos los objetos para su posterior discriminación pero, sin embargo, se complica enormemente el problema. Por este motivo, es necesario reducir el vector de características a una aproximación práctica y por supuesto dependiente del dominio de aplicación.

Cuando se emplea como entrada al sistema de reconocimiento el objeto definido por un vector de características el proceso puede ser entendido como un proceso de clasificación ya que el propósito es asignar el objeto (o patrón) a una de las clases consideradas (o grupos representativos). Cada clase representa un tipo diferente de objeto.

Veamos un ejemplo sencillo de estos conceptos:

Considérese el problema del reconocimiento de tres clases diferentes de objetos: triángulo equilátero, círculo y cuadrado. Para reconocerlos se emplea como única característica la relación entre el perímetro al cuadrado y el área del objeto. Idealmente cada una de estas clases se define mediante unos valores determinados de esta característica (independiente del tamaño), según se recoge en la tabla 5.1.

	ÁREA	PERÍMETRO	$\text{PER}^2/\text{ÁREA}$
círculo	πr^2	$2\pi r$	12'56
cuadrado	l^2	$4l$	16
triángulo	$\frac{\sqrt{3}l^2}{4}$	$3l$	20'8

Tabla 5.1: Valores nominales de la característica para las tres clases

De acuerdo con esta característica, el problema de reconocimiento de un objeto de la imagen como perteneciente a alguna de estas clases podría resolverse, por ejemplo, mediante la definición de una regla de decisión que clasifique según el valor medio de dicha característica. Un simple clasificador sería aquel que divide el espacio de características en los puntos equidistantes de los valores nominales de cada clase. Así, la regla de decisión quedaría:

- si $per^2/área < 14'3$ entonces objeto = círculo

- si $14'3 < per^2/área < 18'4$ entonces objeto = cuadrado
- si $18'4 < per^2/área$ entonces objeto = triángulo

Nótese que, debido a las imprecisiones en los procesos previos a la clasificación (adquisición de imagen, preproceso, segmentación y extracción de características) puede ocurrir que estas fronteras no sean las óptimas o bien que, aunque lo sean produzcan errores en la clasificación. Estas circunstancias hacen pensar en el interés de utilizar en la clasificación información sobre la distribución de las características de cada clase. De esta forma, si éstas presentan una distribución gausiana alrededor de los valores nominales, el grado de la dispersión (dado por la desviación típica) repercutirá en la posición de los valores frontera que particionan el espacio de características de manera óptima.

Tal y como hemos visto el sistema de reconocimiento toma como entrada un patrón y lo asigna a una de las clases según una regla de decisión. La regla de decisión normalmente se obtiene dividiendo el espacio de patrones en zonas disjuntas correspondientes con las clases. Los hiperplanos que separan las clases se denominan fronteras de decisión. La selección de las fronteras de decisión puede realizarse de diversas formas. El método más simple consiste en encontrar la mejor partición del espacio de patrones la cual sitúe todas las muestras disponibles tan lejos como sea posible de la frontera de decisión.

Además puede resultar muy conveniente permitir que las fronteras de decisión se modifiquen a medida que se recibe nueva información dando lugar a un sistema de reconocimiento de patrones adaptativo. En este caso se consideran dos partes diferentes en el problema de reconocimiento: aprendizaje y aplicación. En la fase de aprendizaje las muestras etiquetadas se presentan al sistema secuencialmente y serán empleadas para modificar adecuadamente la regla de decisión con el fin de corregir errores en la clasificación. A este tipo de sistemas en donde el entrenamiento se realiza con muestras etiquetadas se dice que experimenta *aprendizaje supervisado.*

Sin embargo, en muchas ocasiones no se conoce a priori las clases existentes. En

estos casos se emplean muestras no etiquetadas para determinar una clasificación natural para el problema. El método básico en el problema de la clasificación se denomina *clustering* y su objetivo es encontrar subconjuntos de muestras cercanas entre sí. Un clasificador de patrones adaptativo que emplea muestras no etiquetadas se dice que tiene *aprendizaje no supervisado*supervisado. Tales sistemas siguen adaptándose incluso en la fase de aplicación.

Algunos de los métodos existentes para la clasificación de datos con aprendizaje supervisado y no supervisado ya han sido estudiados en el capítulo dedicado a la segmentación.

Hay que dejar claro que el reconocimiento no es posible sin conocimiento. La decisión sobre que clase de objeto está presente en la imagen se basa en el conocimiento de las características que definen cada una de las clases consideradas. Tanto el conocimiento específico sobre los objetos que pueden aparecer en la imagen como el conocimiento general sobre la clase de objetos es necesario utilizar en la etapa de reconocimiento.

5.2 Introducción a la Interpretación

El objetivo de la interpretación de imágenes es la construcción de un modelo interno de la imagen captada por el sensor que represente la imagen procesada del mundo. Esto requiere la iteración de todas las etapas de procesamiento estudiadas en los capítulos anteriores.

Un humano está preparado para procesar, analizar e interpretar una imagen. No obstante puede ocurrir que el humano tenga dificultades en entender una determinada imagen si lo que ve no es conocido. Por ejemplo, si se presenta una imagen del fondo de ojo de un paciente a un observador que no conoce la estructura interna de nuestros ojos, éste no será capaz de detectar si existe o no una patología. Un resultado similar se obtiene si al observador se le requiere interpretar una imagen

previamente captada por un satélite de cierta área urbana aunque dicha zona de la ciudad sea familiar para el observador. Si el observador resuelve el problema de orientación de la escena, se dispone de un punto de partida para intentar localizar un objeto conocido. De este modo, el observador construye un modelo de la ciudad comenzando con el objeto que cree haber identificado inicialmente. Supongamos que la vista de la ciudad es la presentada en la figura 5.1 y supongamos que el observador ha sido capaz de ver una torre. Ésta podría pertenecer a la iglesia o al castillo. El observador comienza con la hipótesis de que la torre detectada pertenece al castillo. Un modelo del castillo consiste de un parque adyacente cercano al río, etc. El observador intenta verificar su hipótesis con el modelo de manera que si la hipótesis corresponde con el modelo implica que es válida. En caso contrario, debe de ser rechazada siendo necesario la construcción de una nueva hipótesis que describa la escena y otro modelo para pasar a continuación a verificarla.

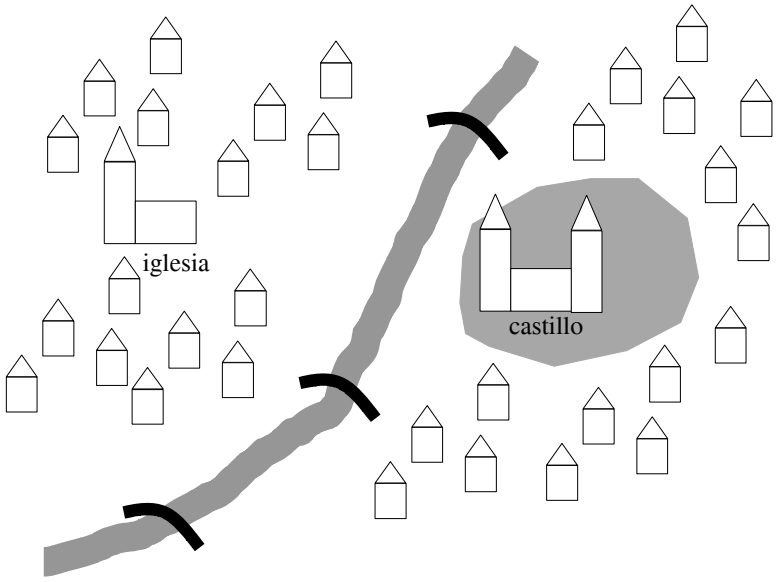


Figura 5.1: Imagen sintética de una ciudad.

Vemos que dos tipos de conocimiento han sido necesarios a lo largo del proceso de interpretación de la imagen: conocimiento general sobre la situación de las calles, parques, casas, etc. en las ciudades y conocimiento específico del orden de las casas

concreto, calles, ríos, etc. de la ciudad captada por el satélite..

Un sistema de visión artificial puede diseñarse para resolver problemas similares. La principal diferencia entre el observador humano y el sistema de visión artificial es la falta de conocimiento modificable, general y ampliamente aplicable del mundo real. Los sistemas de visión artificial construyen modelos internos de la escena procesada, los modifica y los actualiza mediante el desarrollo de una secuencia apropiada de pasos de procesamiento hasta completar la tarea. Si el modelo interno construido corresponde con la realidad implica que la imagen ha sido interpretada correctamente. De esta manera, en el ejemplo descrito anteriormente muestra que el modelo de la imagen es un prerequisito para la percepción. Así, los nuevos datos o percepciones se comparan con el modelo existente y se utilizan para la modificación del modelo obtenido de la imagen.

La interpretación de una imagen no depende únicamente de los datos que la definen. Las variaciones en los modelos de partida así como las diferencia en experiencias previas provocan que los datos se interpreten de forma diferente.

En el primer capítulo diferenciamos entre visión de bajo y alto nivel, estando la interpretación de la imagen en el nivel más alto de la clasificación. Su tarea principal es definir estrategias de control que aseguren una secuencia apropiada de pasos de procesamiento. Además un sistema de visión artificial debe tratar con un gran número de interpretaciones que son hipotéticas y ambiguas. Generalmente, un sistema de visión artificial consiste de una estructura jerárquica de modelos de la imagen.

Un elevado número de resultados importantes en interpretación de imágenes han sido logrados en los últimos años estando, todavía en continua investigación. La interpretación de la imagen es uno de los grandes desafíos de la Inteligencia Artificial e indagar en el área de la visión artificial implica estudiar conceptos relacionados con IA tales como la representación del conocimiento, estructuras relacionales, redes semánticas, correspondencia en general, inferencia, sistemas de producción, resolve-

dores de problemas, planificación, control y aprendizaje a partir de la experiencia. La mayoría de estos conceptos han sido estudiados en la asignatura de Inteligencia Artificial y por supuesto existe numerosa bibliografía dedicada a describirlos. Abordar dichos conceptos en este libro excede sus propósitos por lo que únicamente se presenta una revisión de las estrategias de control para la interpretación de la imagen y se describen métodos semánticos para la interpretación de la imagen. El control de la interpretación de la imagen es un problema crucial en la visión artificial y las estrategias de control descritas ofrecen una forma racional de aplicación de varios de los métodos dedicados al procesado de imagen, segmentación, descripción de objetos y reconocimiento estudiados en los capítulos anteriores.

5.3 Estrategias de control para la interpretación de imágenes

La interpretación de la imagen puede llevarse a cabo como resultado de la cooperación de tareas de procesado de información complejas y el control adecuado de estas tareas. Los sistemas biológicos incluyen estrategias de control complejas incorporando procesamiento en paralelo, modificaciones de comportamiento, interrupción producida por cambios de atención, etc. Como en otros problemas de inteligencia artificial, el principal objetivo de la visión artificial es lograr un comportamiento de la máquina similar al comportamiento biológico mediante la aplicación de procedimientos disponibles técnicamente.

5.3.1 Control de procesos serie y paralelo

Tanto la aproximación serie como paralela puede aplicarse al procesado de la imagen aunque algunas veces no es obvio que paso debe procesarse en paralelo y cual en serie. El procesamiento paralelo realiza varios cómputos simultáneamente (por

ejemplo varias partes de la imagen pueden ser procesadas simultáneamente) siendo extremadamente importante la sincronización de las acciones del procesado, esto es, la decisión de cuando o si el proceso debe esperar a que otros pasos del procesado hayan finalizado.

En el procesado en serie, las operaciones son secuenciales. Una estrategia de control serie es el funcionamiento natural de la arquitectura del computador Von Neumann, sin embargo, la multitud de operaciones que los sistemas biológicos realizan, en general, no puede llevarse a cabo en serie debido principalmente a la velocidad requerida.

Casi todos los procesos de visión de bajo nivel pueden realizarse en paralelo en cambio en la visión de alto nivel, en los niveles más altos de abstracción es normalmente serie. Un observador humano siempre se concentra en un único tópico aunque los pasos previos se realicen de modo paralelo.

5.3.2 Control jerárquico

La información de la imagen se almacena en diferentes representaciones durante el procesado. La cuestión fundamental es que si debe el proceso ser controlado por la información de los datos de la imagen o por el conocimiento del nivel superior. Dependiendo de quien controle el proceso encontramos dos aproximaciones:

1. **Control por los datos de la imagen o control bottom-up**: el procesado está dirigido por los datos de la imagen, es decir, se sigue el orden de las etapas de visión desde el nivel inferior (preproceso) hasta el nivel superior (reconocimiento).

2. **Control basado en el modelo o control top-down**: se define un conjunto de suposiciones y propiedades aplicando el conocimiento. A continuación se comprueba si estas propiedades se satisfacen en le imagen mediante la aplicación de diferentes niveles de procesado desde el nivel superior hasta el inferior

(hasta los datos). La interpretación de la imagen consiste en un proceso de verificación de un modelo el cual puede ser aceptado o rechazado.

Las dos estrategias básicas no difieren en los tipos de operación que pueden ser aplicadas pero si lo hacen en la secuencia de sus operaciones, en su aplicación a todos los datos o únicamente a una parte determinada, etc. El mecanismo de control elegido no es solamente una rutina para alcanzar el objetivo sino que influye en la estrategia de control global. Ni una estrategia ni otra puede explicar el proceso de visión o resolver problemas de visión complejos de manera standard. Sin embargo, su combinación apropiada puede permitirnos definir una estrategia de control más flexible y poderosa.

5.3.3 Estrategias de control Bottom-up

Un algoritmo general bottom-up es:

1. Preprocesado: Permite mejorar los datos de la imagen con el fin de facilitar las tareas siguientes.

2. Segmentación: Detecta las regiones de la imagen que pueden corresponder con objetos reales o partes de los objetos.

3. Extracción de características: Determina una descripción adecuada de las regiones encontradas en la imagen

4. Interpretación: Compara los objetos detectados con los objetos reales considerados en el dominio solución (usando técnicas de reconocimiento de patrones)

La estrategia de control bottom-up presenta ventajas cuando se dispone de un método de procesado eficiente y simple y que sea independiente del contenido de los datos de la imagen. El control bottom-up produce buenos resultados si los datos no presentan ambigüedades y si del procesado se obtienen representaciones precisas

y fiables. Si los datos de entrada son de baja calidad, el control bottom-up puede producir buenos resultados solamente si la no fiabilidad de los datos causa un número limitado de errores insustanciales en cada etapa del proceso. Esto implica que la interpretación de imágenes debe guiarse por una estrategia de control que no sea sólo la concatenación de operaciones de procesado en dirección bottom-up sino que también use un modelo interno, planificación y procesos cognitivos complejos.

5.3.4 Estrategias de control Top-down

No existe una versión específica del control top-down como ha sido definida en el algoritmo de control bottom-up. El principio principal del control top-down es la construcción de un modelo interno y su verificación, esto es, procesado orientado al objetivo. Los objetivos en la visión de alto nivel pueden dividirse en subobjetivos en los niveles de procesado más bajo que a su vez pueden volver a dividirse en otros subobjetivos etc, hasta que los subobjetivos puedan ser aceptados o rechazados directamente.

Veamos un ejemplo para entender esta idea. Desde la ventana de tu casa divisas la zona de aparcamiento donde está aparcado el coche de tu madre (un mercedes A de color rojo) . Imagina que en un momento dado necesitas localizarlo. Lo primero que haces es buscar entre todos los coches aquellos que sean de color rojo. Seguidamente, entre todos los coches rojos seleccionas aquellos que sean mercedes y finalmente de entre ellos buscas los mercedes de clase A. Si todos los objetivos han sido cumplidos, el último de ellos será verificar si el coche detectado es o no el coche de tu madre. Para cumplir este objetivo es necesario disponer de conocimiento específico del coche que buscas. Tienes que saber que diferencia el coche de tu madre del de los demás. Si el test de propiedades específicas del coche detectado se supera con éxito, entonces has localizado el coche de tu madre, el modelo que construiste para el coche es válido pero si el test de propiedades específicas falló debes reanudarlo, por ejemplo detectar otro mercedes A rojo que no haya sido seleccionado.

El mecanismo general del control top-down consiste en la generación secuencial de hipótesis y su evaluación. El modelo interno se actualiza a lo largo del proceso dependiendo de los resultados obtenidos en el test de la hipótesis. La hipótesis que se está evaluando depende de la información adquirida de los niveles de representación más bajos y el control del proceso se basa en el hecho de que para evaluar cada hipótesis se requiere aplicar métodos de procesado de imagen. La estrategia de control basada en modelos parece ser una forma de resolver tareas de visión artificial evitando un alto procesado de la imagen. Al mismo tiempo, no es absurdo pensar en la utilización del procesado paralelo siempre que sea posible.

No sorprende que modelos del mundo reales jueguen un papel sustancial en visión artificial. Para poder modelar objetos complejos es necesario que sus propiedades físicas sean incluidas en la representación. Esto es especialmente verdad en el modelado de objetos naturales. El modelado físico es una rama de la visión artificial en la que se sigue investigando. Las técnicas de modelado pueden incrementar significativamente el conocimiento disponible durante el proceso de interpretación de imágenes.

5.3.5 Estrategias de control combinadas

Un mecanismo de control combinado que emplea las estrategias de control guiadas por los datos y las estrategias de control guiadas por modelo se utiliza ampliamente en aplicaciones de visión artificial. Éstas producen normalmente mejores resultados que cuando se aplican individualmente las estrategias de control básicas. Toda la información de alto nivel disponible se utiliza para hacer el procesado de bajo nivel más fácil pero únicamente ésto sería insuficiente para resolver la tarea. Buscar los coches en una imagen captada por un satélite podría ser un buen ejemplo. El control guiado por los datos es necesario para encontrar los coches pero, al mismo tiempo, el conocimiento de alto nivel puede ser usado para simplificar el problema ya que los coches aparecen como objetos rectangulares de tamaño específico y la mayor

probabilidad de que aparezcan es en carreras.

5.3.6 Control no jerárquico

El control jerárquico puede entenderse como una cooperación de expertos compitiendo en el mismo nivel mientras que en el control jerárquico existen diferentes niveles.

El control jerárquico puede aplicarse a problemas que pueden separarse en un conjunto de subproblemas, cada uno de los cuales requiere un experto. El orden en el cual los expertos han de ser empleados no es fijo. La idea básica del control no jerarquico es pedir asistencia al experto que pueda ayudar a obtener la solución final. El experto elegido podría ser conocido, por ejemplo, por su alta fiabilidad, alta eficiencia, o por su habilidad de dar mayor información en determinadas condiciones., etc. El criterio de selección del experto podría diferir: una posibilidad sería calcular dejar que el experto calcule sus propias habilidades para contribuir a la solución de un caso particular, la elección se basa entonces en estas evaluaciones individuales. Otra opción consiste en asignar una evaluación fija a cada experto de antemano y la ayuda será solicitada al experto con mayor evaluación. El criterio para la selección del experto podría basarse en la combinación apropiada de evaluaciones empíricas calculadas por el experto dependiendo dichas evaluaciones del estado actual de la solución del problema.

Las estrategias de control jerarquico pueden definirse mediante el siguiente algoritmo:

1. Basándonos en el estado actual y en la información adquirida sobre el problema, decidir la mejor acción y ejecutarla.

2. Utilizar los resultados obtenidos en el último paso para aumentar la cantidad de información adquirida sobre el problema.

3. Si se han alcanzado los objetivos de la tarea, parar. En otro caso regresar al paso.

Bibliografía complementaría

1. Capítulo 7 de [10].

2. Capítulo 9 de [4].

3. Capítulo 8 de [5].

4. Capítulo 18 de [3].

5. Capítulos 15, 16, 17, 18, 19 y 20 de [2].

Bibliografía

[1] D. H. Ballard and C. H. Brown. *Computer Vision*. Prentice Hall, 1982.

[2] S. Banks. *Signal Processing Image Processing and Pattern Recognition*. Prentice Hall, 1990.

[3] E. R. Davies. *Machine Vision. Theory. Algorithms. Practicalities*. Academic Press, 1990.

[4] A. De la Escalera. *Visión por computador*. Prentice Hall, 2001.

[5] J. González. *Visión por Computador*. Paraninfo, 1999.

[6] R. C. González and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing Company, 1987.

[7] D. Maravall. *Reconocimiento de formas y visión artificial*. Ra-ma, 1993.

[8] D. Marr. *Vision*. Freeman, New York, 1982.

[9] R. J. Schalkoff. *Digital Processing and Computer Vision. An Introduction to theory and Implementations*. John Wiley Sons, 1989.

[10] M. Sonka, V. Hlavac, and R. Boyle. *Image, Processing, Analysis and Machine Vision*. Chapman Hall, 1993.

[11] P. H. Winston. *The Psychology of Computer Vision*. Mc-Graw Hill, 1975.