

TÉCNICAS Y ALGORITMOS BÁSICOS DE VISIÓN ARTIFICIAL

MATERIAL DIDÁCTICO

Ingenierías

nº 24

Otros títulos de la colección

1. **Planos acotados: expresión gráfica (2ª ed.)**
Ricardo Bartolomé Ramírez
2003, 306 pags. ISBN 84-95301-74-1
2. **Lenguaje “Turbo C” para estudiantes**
Francisco J. Martínez de Pisón
1996, 191 pags. ISBN 84-88713-33-9
3. **Problemas de análisis de circuitos eléctricos. Corriente continua. Corriente alterna monofásica**
Montserrat Mendoza Villena, Luis Alfredo Fernández Jiménez.
1997, 142 pags. ISBN 84-88713-58-4
4. **Problemas de electrónica analógica**
Antonio Zorzano Martínez
1999, 118 pags. ISBN 84-88713-96-7
5. **Programar es fácil**
Julio Blanco Fernández
1999, 250 pags. ISBN 84-88713-97-5
6. **Problemas resueltos de topografía práctica**
Jacinto Santamaría Peña
1999, 84 pags. ISBN 84-88713-98-3
7. **Resistencia de materiales. Nivel básico**
Eduardo Martínez de Pisón Ascacibar
1999, 316 pags. ISBN 84-95301-14-8
8. **Prácticas de C.A.D. Microstation 2D (2ª ed.)**
José Lafargue Izquierdo
1999, 224 pags. ISBN 84-95301-15-6
9. **Programación de proyectos**
Joaquín Ordieres Meré
1999, 96 pags. ISBN 84-95301-16-4
10. **Termodinámica fundamental (2ª ed.)**
J. M. Sala Lizarraga, Luis M. López
2000, 448 pags. ISBN 84-95301-25-3
11. **Termodinámica aplicada (2ª ed.)**
J. M. Sala Lizarraga, L. M. López y Víctor de la Peña
2000, 584 pags. ISBN 84-95301-26-1
12. **Problemas Termodinámica fundamental (2ª ed.)**
J. M. Sala Lizarraga, Luis M. López y Felipe Jiménez
2000, 490 pags. ISBN 84-95301-27-X
13. **Problemas Termodinámica aplicada (2ª ed.)**
J. M. Sala Lizarraga, Luis M. López y M.M. Ruiz de Adana
2000, 432 pags. ISBN 84-95301-28-8
14. **Problemas de calor y frío industrial**
L. M. López, J. M. Sala y J. M. Blanco Ilzarbe
2000, 418 pags. ISBN 84-95301-29-6
15. **Apuntes de cartografía y proyecciones cartográficas**
Jacinto Santamaría Peña
2000, 74pags. ISBN 84-95301-30-X
16. **Apuntes de fotogrametría**
Jacinto Santamaría Peña y Teófilo Sanz Méndez
2000, 68pags. ISBN 84-95301-30-X
17. **Perspectiva: fundamentos y aplicaciones. Axonométrico. Caballera. Cónico**
Ricardo Bartolomé Ramírez
2000, 260 pags. ISBN 84-95301-33-4
18. **Problemas de resistencia de materiales. Nivel básico. Ingeniería agrícola**
Eduardo Martínez de Pisón Ascacibar
2001, 446 pags. ISBN 84-95301-44-X
19. **Sonometría y contaminación acústica.**
Javier de Cos, J. Ordieres, M. Castejón, F. J. Martínez de Pisón
2001, 384 pags. ISBN 84-95301-47-4
20. **Cuadernos de prácticas de informática industrial. Modulo 1: enunciados de prácticas en ensamblador**
F. J. Martínez de Pisón, J. Ordieres, M. Castejón, F. J. de Cos, M. Gil.
2001, 110 pags. ISBN 84-95301-58-X
21. **La oficina técnica y los proyectos industriales**
F. J. Martínez de Pisón, J. Ordieres, M. Castejón, F. J. de Cos, E. P. Vergara, F. Alba.
2 v. ISBN 84-95475-32-4
22. **Manual de prácticas de topografía y cartografía**
Jacinto Santamaría Peña.
Edición electrónica
115 págs. ISBN 84-689-4103-4
23. **Problemas de electrotecnia**
Edición electrónica.
José Fernando Azofra Catroviejo
113 págs. ISBN 84-689-7232-0

Ana González Marcos
Francisco Javier Martínez de Pisón Ascacibar
Alpha Verónica Pernía Espinoza
Fernando Alba Elías
Manuel Castejón Limas
Joaquín Ordieres Meré
Eliseo Vergara González
(Integrantes del Grupo de Investigación EDMANS)

TÉCNICAS Y ALGORITMOS BÁSICOS DE VISIÓN ARTIFICIAL

UNIVERSIDAD DE LA RIOJA
SERVICIO DE PUBLICACIONES
2006

Técnicas y algoritmos básicos de visión artificial / Ana González Marcos... [et al.] (integrantes del Grupo de Investigación EDMANS). – [Logroño] : Universidad de La Rioja, Servicio de Publicaciones, 2006
XXX p. : il. ; 30 cm. – (Materia didáctico. Ingenierías ; 24)
ISBN 84-689-9345-X
1. Visión por ordenador. I. González Marcos, Ana. II. Universidad de La Rioja. Grupo de Investigación EDMANS
681.31.019

Reservados todos los derechos. No está permitida la reproducción total o parcial de este libro, bajo ninguna forma ni por ningún medio, electrónico o mecánico, ni por fotocopia o grabación, ni por ningún otro sistema de almacenamiento, sin el permiso previo y por escrito de los titulares del Copyright.

© Ana González Marcos, Francisco Javier Martínez de Pisón Ascacibar, Alpha Verónica Pernía Espinoza, Fernando Alba Elías, Manuel Castejón Limas, Joaquín Ordieres Meré y Eliseo Vergara González (Integrantes del Grupo de Investigación EDMANS)
Universidad de La Rioja. Servicio de Publicaciones

Edita: Universidad de La Rioja. Servicio de Publicaciones
Diseño de portada: Universidad de La Rioja. Servicio de Comunicación

ISBN 84-689-9345-X

Impreso en España - Printed in Spain

ÍNDICE

1. INTRODUCCIÓN.....	11
1.1 Fundamentos	11
1.1.1 Formación y representación de la imagen.....	11
1.1.2 Efectos del muestreo y la cuantificación.....	12
1.1.3 Relaciones básicas entre píxels.....	13
1.2 Etapas en un proceso de visión artificial.....	15
1.3 Componentes de un sistema de visión.....	16
1.4 Introducción a las aplicaciones.....	17
2. ADQUISICIÓN DE IMÁGENES.....	19
2.1 La Cámara	19
2.1.1 Conclusiones para la selección de la cámara	20
2.2 La óptica.....	20
2.2.1 Conclusiones para la selección de la óptica.....	21
2.3 La tarjeta de adquisición de imagen.....	21
2.3.1 Conclusiones para la selección de la tarjeta de visión.....	23
2.4 El <i>host</i> y el software utilizado	23
2.4.1 Selección del tipo de <i>host</i>	23
2.4.2 Elección del software de programación y visualización.....	24
2.4.3 Conclusiones para la selección del ordenador y del software adecuado.....	25
2.5 Técnicas de iluminación	25
2.5.1 La iluminación	26
2.5.2 El Fondo.....	29
2.5.3 Conclusiones para la selección de la iluminación y del fondo.....	29
2.6 Calibración	30
2.6.1 Conclusiones para realizar un calibrado correcto	31
3. PREPROCESADO.....	33
3.1 Operaciones básicas.....	33
3.1.1 Inversión.....	33
3.1.2 Operaciones Aritméticas.....	34
3.1.3 Operaciones lógicas.....	34
3.1.4 Transformaciones no lineales.....	35
3.1.5 <i>Slicing</i>	35
3.1.6 <i>Clipping</i>	36
3.1.7 Umbralización.....	36
3.1.8 Binarización	36
3.1.9 Otras funciones	36
3.2 Transformaciones geométricas de imágenes	36
3.2.1 Algoritmos de traslación o desplazamiento.....	37
3.2.2 Algoritmos de rotación o giro	37
3.2.3 Algoritmos de escalado y de <i>zoom</i>	37
3.2.4 Espejos.....	37
3.2.5 Algoritmos que reducen la resolución de la imagen	38
3.2.6 Deformación.....	38
3.2.7 Otras transformaciones geométricas.....	38
3.3 Algoritmos generadores de ruido.....	39
3.3.1 Generadores de ruido dependiente de la señal	39
3.3.2 Generadores de ruido aleatorio	39
3.4 Filtrado	39
3.4.1 Filtros frecuenciales.....	40
3.4.2 Filtros espaciales	42
3.4.3 Otros algoritmos de filtrado.....	49

3.5 Operaciones basadas en histograma	49
3.5.1 Transformaciones del histograma	50
3.5.2 Ecuación del histograma	52
4. SEGMENTACIÓN	55
4.1 Transformaciones morfológicas	55
4.1.1 Dilatación binaria	55
4.1.2 Erosión binaria	57
4.1.3 Apertura	58
4.1.4 Cierre	58
4.1.5 Esqueletos	60
4.1.6 Otras transformaciones	60
4.2 Técnicas de segmentación	61
4.2.1 Segmentación basada en umbralización	61
4.2.2 Técnicas de segmentación basadas en bordes	64
4.2.3 Segmentación orientada a regiones	69
4.2.4 Técnicas de clusterizado	72
4.2.5 Segmentación por moción	72
4.2.6 Otras técnicas de segmentación	72
5. ANÁLISIS	73
5.1 Representaciones de las fronteras	73
5.1.1 Códigos de cadena	73
5.1.2 Signaturas	74
5.1.3 Aproximaciones poligonales	74
5.2 Representaciones de las regiones	75
5.2.1 Códigos de longitud variable	75
5.2.2 Quad-trees	75
5.2.3 Proyecciones	76
5.3 Descriptores de frontera	76
5.3.1 Longitud	77
5.3.2 Diámetro	77
5.3.3 Curvatura	77
5.3.4 Descriptores de Fourier	77
5.4 Descriptores de región	78
5.4.1 Medida de área	78
5.4.2 Centro de gravedad	79
5.4.3 Caja del contorno	79
5.4.4 Momentos centrales	80
5.4.5 Cuerdas	81
5.4.6 Perímetro	82
5.4.7 Diámetros de Feret	83
5.4.8 Puntos significativos del contorno	84
5.4.9 Ejes	84
5.4.10 Orientación	85
5.4.11 Medidas de distancia polar	85
5.4.12 Rugosidad	86
5.4.13 Redondez	86
5.4.14 Número de huecos	87
5.4.15 Número de Euler	87
5.5 Medidas Densitométricas	87
5.5.1 Función de autocorrelación	87
5.5.2 Energía	87
5.5.3 Parámetros ópticos	88
5.5.4 Parámetros estadísticos	88
5.5.5 Descripción de texturas	89
5.6 Medidas de líneas y puntos	90
5.6.1 Medidas de líneas geométricas	90

PRÓLOGO

El libro titulado “Técnicas y Algoritmos Básicos de Visión Artificial” surge de las experiencias obtenidas, en estos últimos años, de varios Proyectos de Visión Artificial que han realizado los autores en varias empresas del sector Agroalimentario e Industrial. Éste libro, pretende ser una pequeña recopilación de aquellas técnicas y algoritmos que consideramos han resultado más efectivos, fundamentalmente por su sencillez y eficacia.

El libro está dividido en cinco capítulos. En los dos primeros se realiza una breve introducción de los sistemas de visión artificial y de la importancia de factores como la iluminación, el control del entorno, las cámaras, el hardware, etc. En el capítulo tercero, se abordan los algoritmos de preprocesado, fundamentales para el tratamiento previo de la imagen. En el capítulo cuarto se exponen algunos algoritmos y técnicas utilizados en las tareas de extracción o segmentación de los objetos de la imagen. Por último, en el capítulo quinto se plantean aquellas técnicas que permiten parametrizar los objetos segmentados.

Nuestro máximo interés, por lo tanto, consiste en poder presentar una pequeña guía que pueda ser de ayuda a los estudiantes que realicen Proyectos Fin de Carrera o estudien asignaturas afines, así como para Investigadores o Profesionales que deseen adquirir conocimientos básicos de dichas técnicas.

LOS AUTORES

1. INTRODUCCIÓN

La ciencia informática camina actualmente hacia la creación de computadoras cada vez más rápidas, más expertas y más autónomas. Una de las empresas más ambiciosas consiste en dotar a los ordenadores de la facultad de relacionarse con su entorno del mismo modo que lo hace un humano: a través de los sentidos.

Proporcionar capacidad sensorial a un ordenador es una tarea difícil ya que entran en juego elementos accesorios al microprocesador, como son los sensores, las tarjetas adaptadoras de señal o el ruido del entorno. A pesar de los muchos inconvenientes, existe hoy un interés especial por dotar a los ordenadores de uno de los cinco sentidos del hombre: la habilidad de ver. Mientras que el resto de los sentidos humanos no despierta interés, la visión artificial se aplica ya en procesos industriales en los que la simple detección de presencia no resulta una fuente de información suficiente.

La visión artificial es una disciplina que engloba todos los procesos y elementos que proporcionan ojos a una máquina y se podría decir que: *la visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo*¹. Las estructuras y propiedades del mundo tridimensional que se quieren deducir en visión artificial incluyen no sólo sus propiedades geométricas, sino también sus propiedades materiales. Ejemplos de propiedades geométricas son la forma, tamaño y localización de los objetos. Ejemplos de propiedades de los materiales son su color, iluminación, textura y composición. Si el mundo se modifica en el proceso de formación de la imagen, se necesitará inferir también la naturaleza del cambio, e incluso predecir el futuro.

La visión, tanto para un hombre como para un ordenador, consta principalmente de dos fases: captar una imagen e interpretarla. A pesar de la complejidad que presenta el ojo humano, la fase de captación de imágenes hace mucho tiempo que está resuelta. El *ojo* del ordenador es la cámara de vídeo, y su *retina* un sensor que es sensible a la intensidad luminosa. Así que en la visión artificial lo que resta es interpretar las imágenes, distinguir los objetos de la escena, extraer información de ellos y resolver aspectos más particulares según las necesidades que se deseen satisfacer.

La interpretación de tramas complejas, sin embargo, es una capacidad reservada al cerebro humano. Para que un ordenador fuera capaz de analizar directamente una imagen *cruda*, sin procesar, haría falta una estructura de procesadores a modo de neuronas, además de una ingente cantidad de información de los objetos y las formas. Para evitarlo, lo que se hace es reducir la complejidad de la trama. Para el ordenador resultará más difícil distinguir objetos que se encuentren sobre un fondo con una trama del mismo tono y proyecten una sombra definida, que reconocerlos sobre un fondo plano que contraste con su color frontal. Es por ello que una de las más amplias ramas de la visión artificial está guiada precisamente a simplificar las imágenes, eliminar el ruido introducido por los sensores y facilitar, en definitiva, la tarea de interpretación.

Las tareas de preprocesado consisten en algoritmos matemáticos que calculan nuevas intensidades luminosas para los pixels y la mayoría, aunque sencillos, consumen gran cantidad de tiempo de cálculo, tanto mayor cuanto más resolución tenga la imagen. Esto representa un inconveniente cuando se pretende que el ordenador sea capaz de ver en tiempo real, es decir, que responda de forma casi instantánea a como se producen las imágenes frente a sus *ojos*. Para que un sistema funcione en tiempo real es necesario estudiar todos los métodos posibles y la forma de realizarlos con la menor carga computacional posible.

Actualmente la visión computacional está en fase de crecimiento. Dada la enorme complejidad de ésta, se van solucionando, por etapas, problemas cada vez más complicados. Actualmente, las líneas de estudio e investigación se dividen en múltiples campos, desde la visión tridimensional y visión con moción, a la segmentación y agrupamiento de múltiples objetos diferentes en entornos no controlados, etc..

1.1 FUNDAMENTOS

1.1.1 FORMACIÓN Y REPRESENTACIÓN DE LA IMAGEN

Una imagen es una representación visual de un objeto iluminado por una fuente radiante. Las que se perciben en las actividades visuales cotidianas provienen normalmente de la luz reflejada por los objetos. La naturaleza básica de una imagen, representada por $f(x,y)$ ², está caracterizada por dos componentes: la cantidad de luz incidente que

¹ V.S. Nalwa. *A Guided Tour to Computer Vision*. Addison-Wesley, 1993.

² A lo largo de este documento se usarán normalmente $f(x,y)$ o $g(x,y)$ para notar una imagen. Para pixels particulares se emplearán letras minúsculas como p y q . Un subconjunto de pixels de $f(x,y)$ se nota S .

procede de la fuente de la escena contemplada; y la cantidad de luz reflejada por los objetos de la escena. Dichas componentes reciben el nombre de iluminación y reflectancia, notándose $i(x,y)$ y $r(x,y)$ respectivamente. Ambas funciones se combinan como producto para dar $f(x,y)$.

En el proceso de formación de la imagen intervienen los siguientes elementos: el objeto, la fuente radiante y el sistema de formación de la imagen que consiste, básicamente, en un sistema óptico, un sensor y un digitalizador.

La imagen digital puede ser representada por una matriz f de dimensiones $N \times M$ de la forma:

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,M) \\ f(2,1) & f(2,2) & \cdots & f(2,M) \\ \vdots & \vdots & \cdots & \vdots \\ f(N,1) & f(N,2) & \cdots & f(N,M) \end{bmatrix}$$

donde cada elemento, pixel, da la intensidad de la imagen en ese punto.

1.1.2 EFECTOS DEL MUESTREO Y LA CUANTIFICACIÓN

Las dos principales causas que producen pérdida de información cuando se captura una imagen son la naturaleza discreta de los pixels de la imagen y el rango limitado de valores de intensidad luminosa que es capaz de ofrecer el sistema de digitalización.

EFECTO DEL MUESTREO

El muestreo de una imagen tiene el efecto de reducir la resolución espacial de la misma. La Figura 1.1 muestra el efecto de captar una misma imagen a distintas resoluciones. Todas ellas han sido transformadas al mismo tamaño para su visualización.

Puede observarse la pérdida de información introducida con el aumento del paso de muestreo, así como el ruido que se va introduciendo en forma de patrones rectangulares sobre la imagen.

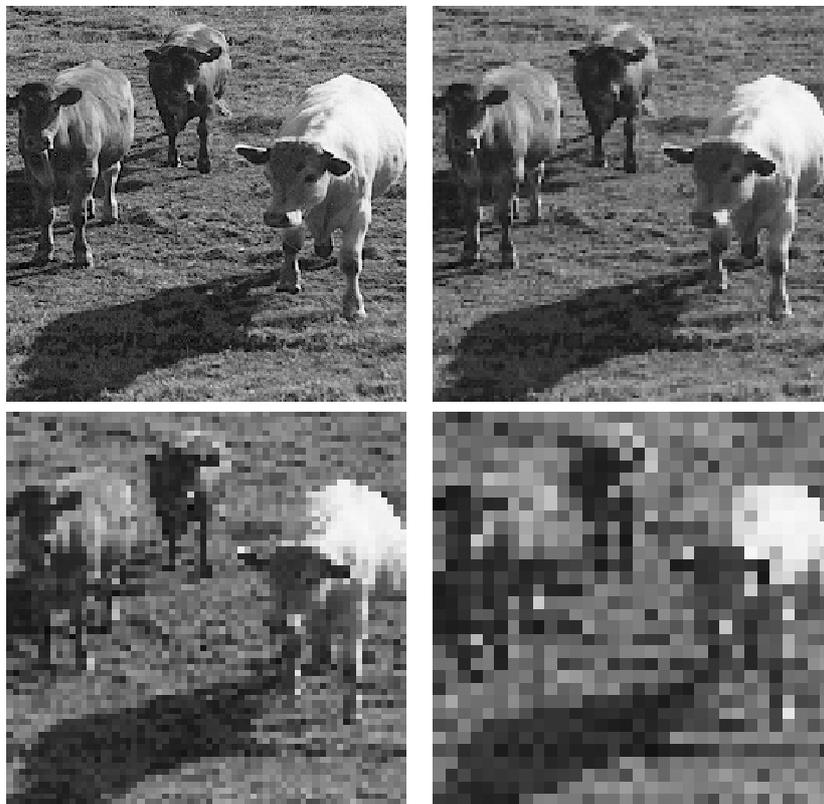


Figura 1.1. De izquierda a derecha y de arriba abajo: muestreo a 256x256, 128x128, 64x64 y 32x32 pixels.

EFFECTO DE LA CUANTIFICACIÓN

El efecto de cuantificación viene dado por la imposibilidad de tener un rango infinito de valores de medida para la intensidad de brillo de los pixels. La tecnología actual permite en algunos casos llegar hasta 10 bits de información, aunque lo general es tener 8 bits, o equivalentemente 256 niveles de gris para codificar este valor lumínico. La Figura 1.2 muestra el efecto de representar una imagen con distintos números de bits.



Figura 1.2. Efecto de la cuantificación. De izquierda a derecha: 8 y 1 bits

No existen criterios que permitan decidir el número óptimo de pixels y de bits con los que muestrear una determinada imagen. Distintos estudios experimentales, usando la opinión de expertos, han llegado a la conclusión de que existen determinados valores umbral por encima de los cuales no se aprecia una significativa ganancia pero por debajo de ellos sí se aprecia una pérdida efectiva en la imagen. En cualquier caso y dado el estado de desarrollo actual de los métodos automáticos de análisis e interpretación de imágenes la conclusión de obtener una imagen con la mayor resolución y calidad posible parece por ahora la única viable.

1.1.3 RELACIONES BÁSICAS ENTRE PIXELS

En esta sección se van a considerar varias relaciones simples aunque importantes entre los pixels de una imagen digital.

VECINOS DE UN PIXEL

Un pixel p de coordenadas (x, y) tiene cuatro vecinos horizontales y verticales cuyas coordenadas son

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \quad (1.1)$$

Este conjunto de pixels, que recibe el nombre de 4-vecinos de p , se nota $N_4(p)$. Cada pixel está a distancia unitaria de (x, y) , y algunos de los vecinos de p caen fuera de la imagen digital si (x, y) está en el borde de la imagen.

Los pixels diagonales vecinos de p tienen coordenadas

$$(x + 1, y + 1), (x - 1, y - 1), (x - 1, y + 1), (x + 1, y - 1) \quad (1.2)$$

y se notan $N_D(p)$. Estos puntos, junto con los cuatro vecinos, se llaman 8 - vecinos de p y se notan $N_8(p)$. Al igual que antes, algunos puntos de $N_D(p)$ y $N_8(p)$ caen fuera de la imagen si (x, y) está en el borde de la misma.

CONECTIVIDAD

La conectividad entre pixels es un concepto importante usado para establecer las fronteras de objetos y las regiones componentes de una imagen. Para establecer si dos pixels están conectados debe determinarse si son adyacentes en algún sentido (por ejemplo si son 4-vecinos y si sus niveles de gris cumplen algún criterio de similitud, por ejemplo ser iguales). Así o en una imagen binaria con valores 0 y 1 dos pixels pueden ser 4-vecinos y no estar conectados salvo que tengan el mismo valor.

Sea V el conjunto de valores de niveles de gris que se usa para definir la conectividad; por ejemplo en las imágenes binarias, $V = \{1\}$ o un rango como $V = \{33, 34, \dots, 50\}$ en las imágenes de niveles de gris.

Se consideran tres tipos de conectividad:

4 - conectividad. Dos pixels p y q con valores en V se dicen 4-conectados si q pertenece a $N_4(p)$.

8-conectividad. Dos pixels p y q con valores en V se dicen 8-conectados si q pertenece a $N_8(p)$.

m-conectividad (conectividad mixta). Dos pixels p y q con valores en V se dicen m-conectados si $q \in N_4(p)$ o

$q \in N_D(p)$ y $N_4(p) \cup N_4(q)$ es vacío. (Este es el conjunto de 4-vecinos de p y q con valores en V).

Es importante notar que la m-conectividad se introduce para eliminar la ambigüedad en los posibles caminos que unen dos pixels.

Se dice que un pixel p es adyacente a q si están conectados. Es obvio que este concepto depende del tipo de conectividad que se use. Dos subconjuntos de una imagen S , S_1 y S_2 , son adyacentes si algún pixel en S_1 es adyacente a alguno en S_2 .

Un camino del pixel p con coordenadas (x, y) a q con coordenadas (s, t) es una sucesión distinta de pixels con coordenadas $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$; donde $(x_0, y_0) = (x, y)$ y $(x_n, y_n) = (s, t)$, siendo (x_i, y_i) adyacente a (x_{i-1}, y_{i-1}) para $1 \leq i \leq n$, siendo entonces n la longitud del camino. Es obvio que el tipo de camino depende del tipo de adyacencia utilizado.

Si p y q son pixels en un subconjunto S de una imagen, entonces p está conectado a q en S si existe un camino de p a q que esté contenido en S .

MEDIDAS DE DISTANCIA

Para los pixels p, q y z con coordenadas $(x, y), (s, t)$ y (u, v) respectivamente, D es una función de distancia o métrica si cumple:

$$D(p, q) \geq 0 \quad (D(p, q) = 0 \text{ si y sólo si } p = q)$$

$$D(p, q) = D(q, p) \text{ y}$$

$$D(p, z) \leq D(p, q) + D(q, z)$$

La distancia euclídea entre p y q se define como:

$$D_e(p, q) = \sqrt{(x - s)^2 + (y - t)^2} \quad (1.3)$$

Para esta medida de distancia, los pixels a distancia menor o igual que un valor r de uno dado, (x, y) , definen un disco de radio r centrado en él.

La distancia D_4 (también llamada de bloques por la estructura que genera) se define mediante:

$$D_4(p, q) = |x - s| + |y - t| \quad (1.4)$$

En este caso, los pixels que estén a una distancia, de (x, y) , D_4 menor o igual que un determinado valor r forman un rombo centrado en (x, y) . Por ejemplo, los pixels a distancia menor o igual que 2 de uno dado son los siguientes:

$$\begin{array}{ccccccc} & & & & & & 2 \\ & & & & & & 2 \ 1 \ 2 \\ & & & & & & 2 \ 1 \ 0 \ 1 \ 2 \\ & & & & & & 2 \ 1 \ 2 \\ & & & & & & 2 \end{array}$$

La distancia D_8 (también llamada de ajedrez) entre p y q se define mediante:

$$D_8(p, q) = \max(|x - s|, |y - t|) \quad (1.5)$$

Por ejemplo los pixels con distancia D_8 menor o igual que 2 son:

```

2 2 2 2 2
2 1 1 1 2
2 1 0 1 2
2 1 1 1 2
2 2 2 2 2
    
```

1.2 ETAPAS EN UN PROCESO DE VISIÓN ARTIFICIAL

La visión artificial lleva asociada una enorme cantidad de conceptos relacionados con hardware, software y también con desarrollos teóricos. En esta sección se verán los pasos fundamentales, recogidos en la Figura 1.3, para llevar a cabo una tarea de visión artificial.

El primer paso en el proceso es adquirir la imagen digital. Para ello se necesitan sensores y la capacidad para digitalizar la señal producida por el sensor.

Una vez que la imagen digitalizada ha sido obtenida, el siguiente paso consiste en el preprocesamiento de dicha imagen. El objetivo del preprocesamiento es mejorar la imagen de forma que el objetivo final tenga mayores posibilidades de éxito.

El paso siguiente es la segmentación. Definida en sentido amplio, su objetivo es dividir la imagen en las partes que la constituyen o los objetos que la forman. En general la segmentación autónoma es uno de los problemas más difíciles en el procesamiento de la imagen. Por una parte, una buena segmentación facilitará mucho la solución del problema; por otra, la segmentación errónea conducirá al fallo.

La salida del proceso de segmentación es una imagen de datos que, o bien contienen la frontera de la región o los puntos de ella misma. Es necesario convertir estos datos a una forma que sea apropiada para el ordenador. La primera decisión es saber si se va a usar la representación por frontera o región completa. La representación por la frontera es apropiada cuando el objetivo se centra en las características de la forma externa como esquinas o concavidades y convexidades. La representación por regiones es apropiada cuando la atención se centra en propiedades internas como la textura o el esqueleto. Sin embargo, en muchas aplicaciones ambas representaciones coexisten.

La elección de una representación es sólo una parte de la transformación de los datos de entrada. Es necesario especificar un método que extraiga los datos de interés. La parametrización, que recibe también el nombre de selección de rasgos se dedica a extraer rasgos que producen alguna información cuantitativa de interés o rasgos que son básicos para diferenciar una clase de objetos de otra.

En último lugar se encuentran el reconocimiento y la interpretación. El reconocimiento es el proceso que asigna una etiqueta a un objeto basada en la información que proporcionan los descriptores (clasificación). La interpretación lleva a asignar significado al conjunto de objetos reconocidos.

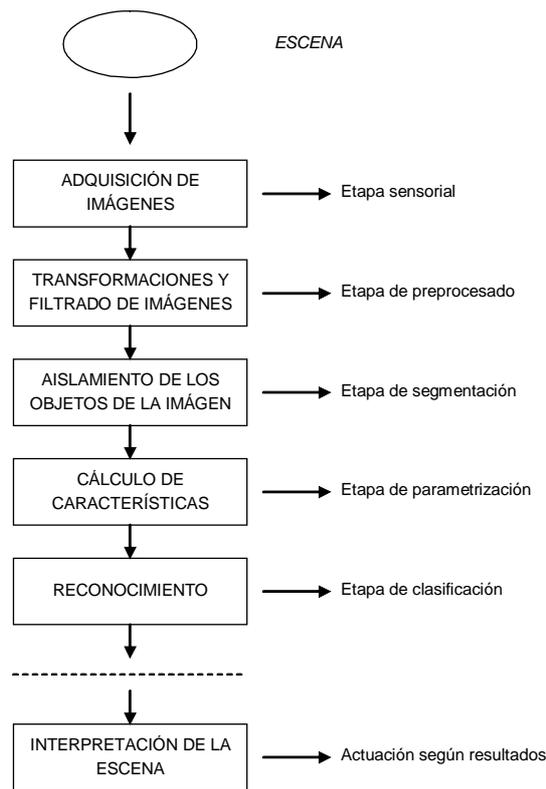


Figura 1.3. Diagrama de bloques de las etapas de un sistema de visión artificial.

1.3 COMPONENTES DE UN SISTEMA DE VISIÓN

Los elementos de hardware mínimos necesarios para un sistema de visión artificial (SVA) son los siguientes:

Sensor óptico. El sensor puede ser una cámara color o monocromo que produce una imagen completa del dominio del problema cada 1/30 segundos. Los sensores podrían ser también una cámara scanner que produce una línea en cada instante. En este caso el movimiento del objeto por la línea del scanner (o al revés) produce la imagen bidimensional. La naturaleza del sensor y la imagen que produce vienen determinadas por la aplicación.

Tarjeta de adquisición de imagen. Permite digitalizar la señal de video entregada por el subsistema anterior.

Computador. Una vez digitalizada la imagen, debe ser almacenada en la memoria de un computador para su posterior procesamiento y manipulación por programa.

Monitor de vídeo. Permite visualizar tanto las imágenes o escenas captadas como los resultados del procesamiento de dichas imágenes.

En la Figura 1.4 se pueden ver el objeto observado, la iluminación, la cámara y el proceso de digitalización que permite por un lado la visualización de la imagen digital y por otro lado el proceso de extracción de parámetros y su posterior proceso de análisis y control que actúa sobre el entorno del objeto, ya sea para: clasificarlo, desestimarlo, actuar sobre él, etc.

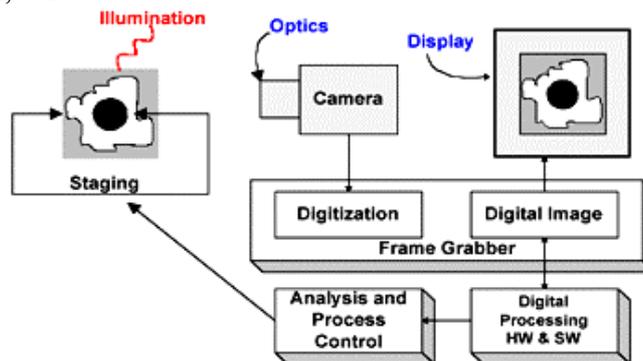


Figura 1.4. Diagrama de bloques de un sistema SVA.

1.4 INTRODUCCIÓN A LAS APLICACIONES

Es obvio que la visión artificial tiene varias aplicaciones menos ambiciosas que la construcción de robots autónomos. Estas aplicaciones pueden pasar por la automatización de una línea de ensamblaje, la inspección de circuitos para detectar defectos, teledetección para construir mapas de regiones no fácilmente accesibles, comunicación mediante gestos con el ordenador y ayudas a los discapacitados mediante perros guías robots,

Entre los sectores donde más se están utilizando actualmente la tecnología de visión artificial se encuentran las industrias: auxiliar del automóvil, farmacéutica, embotellado y embalaje, etiquetado, textil, papel, metalúrgica, semiconductores, cerámica, madera, equipos electrónicos, etc..

2. ADQUISICIÓN DE IMÁGENES

La primera etapa, dentro de un proceso de visión computacional es la **etapa de adquisición**. En este primer paso, se trata de conseguir que la imagen sea lo más adecuada posible para que se pueda continuar con las siguientes etapas.

Una correcta adquisición de la imagen supone un paso muy importante para que el proceso de reconocimiento tenga éxito. Dentro de esta etapa existen múltiples factores que atañen directamente al proceso de captura de la imagen, formados fundamentalmente por: **el sistema hardware de visión artificial** (cámara, óptica, tarjeta de adquisición de imagen, ordenador y software) y **el entorno y posicionamiento de los elementos** (la iluminación, el fondo, posición correcta de la cámara, ruido eléctrico-óptico externo, etc.).

2.1 LA CÁMARA

Lo primero que hay que elegir, a la hora de diseñar un SVA, es el tipo de cámara a utilizar. La elección va a depender de múltiples factores que se detallan a continuación:

Cámara de color o monocromática: Lo primero que debemos de decidir es si utilizamos una cámara de color o de escala de grises. Generalmente, además del presupuesto que tengamos ya que una cámara color es mucho más cara, se elegirá dependiendo del proceso que vayamos a realizar. La mayoría de los casos en los que se utiliza un SVA (dentro del entorno industrial) la adquisición se realiza **mediante una cámara monocromática**, siendo las ventajas principales económicas y de cómputo. Hay que darse cuenta que una imagen en color necesita mucho más tiempo de procesado y para procesos en tiempo real la velocidad de cálculo es un parámetro importante a considerar. Por otra parte, la mayoría de los procesos se basan en el reconocimiento de formas a partir de la determinación de los contornos de los objetos, siendo innecesaria la información cromática de los objetos. De todas formas existen aplicaciones en las que la información de color es fundamental, por ejemplo en control de calidad en la industria alimentaria donde se clasifica la fruta, verdura, etc.; según el color de cada elemento; o por ejemplo en la determinación del tipo de acero fundido o su temperatura según el color que irradia, o en piezas de diferentes colores, etc. La imagen en color realmente proporciona una información más completa que puede servir en la etapa de segmentación y extracción de parámetros en detrimento de una mayor necesidad de tiempo de cómputo.

Cámara digital o analógica: Actualmente empiezan a aparecer en el mercado las cámaras digitales, las ventajas de estas cámaras respecto a las analógicas estriba en su velocidad y calidad de la imagen que presentan. Las cámaras digitales transmiten la información directamente en digital, lo que supone una mejor calidad frente a ruidos que puedan aparecer en los elementos de transmisión (cables, conectores, etc.) además pueden implementar bits de detección/corrección de errores que permitan verificar y corregir la información recibida, como desventaja el cable necesario es de un espesor considerable (formado por gran cantidad de hilos). Desgraciadamente existen pocas cámaras y tarjetas en el mercado además de que su precio es considerablemente superior. Especial cuidado hay que tomar en la construcción del cable, sea éste para tarjeta digital o analógica, realizando un buen apantallado para evitar ruido en la imagen capturada.

Tipo de salida de vídeo: Actualmente podemos encontrar tres tipos de salida de vídeo: vídeo compuesto, salida de vídeo digital y salida RGB (para cámaras color). Las ventajas de una cámara de salida digital se han explicado en el punto anterior, por otro lado las cámaras de salida RGB, tienen los tres colores fundamentales separados, lo que permite poder actuar sobre cada color directamente desde el hardware; de todas formas no se desprenden más ventajas.

Formato PAL o NTSC: Existen principalmente dos formatos de vídeo: el formato europeo PAL (50 Hz) o el americano NTSC (60 Hz). Existen cámaras que tiene uno u otro formato. Hay que verificar que la tarjeta permite capturar imágenes con el formato de la cámara que se tenga.

Salida entrelazada o no entrelazada: Nos podemos encontrar con cámaras de vídeo con salida entrelazada, no entrelazada o que permiten elegir una u otra. La salida entrelazada es adecuada para sacarla directamente por un monitor entrelazado y es muy utilizada para todas las aplicaciones donde se estudian objetos quietos o que se desplazan de una forma lenta, mientras que **la salida no entrelazada es más adecuada para la adquisición de imágenes en movimiento**. Lo ideal es tener una cámara que permita las dos formas.

Sincronismos Vertical y Horizontal: Una buena cámara tiene que permitir el que se le puedan introducir externamente los sincronismos horizontal y vertical, además de que se puedan extraer sus propios sincronismos internos. La salida o entrada de sincronismos permite una completa iteración entre la tarjeta y la cámara, **puediéndose**

realizar la captura de los frames en el momento que deseamos. Pueden existir eventos que se producen de manera muy rápida, siendo fundamental el uso de los sincronismos o también el de los *triggers* que tenga la cámara para la captura correcta de la imagen que deseamos. Dentro de la salida en RGB, generalmente **las cámaras de este formato suelen tener el sincronismo en la salida G**, aunque existen cámaras que tienen la posibilidad de sacar el sincronismo por otra salida aparte.

Velocidad del obturador: Para poder obtener imágenes en movimiento, es necesario controlar el tiempo de exposición. Una cámara que se precie tiene que tener la posibilidad de programarse la velocidad de su obturador (ya sea mediante software o mediante *jumpers*). Generalmente existen cámaras con tiempos de 1/60, 1/125, 1/250, 1/500, 1/1000, 1/2000, 1/4000, 1/10000 segundos de obturación.

Características del CCD: El CCD es una de las partes más importantes de la cámara, dentro de la elección de ésta no se debe de dejar de considerar las características fundamentales de su CCD:

Resolución: Cuanto más resolución tiene una cámara mejor podemos distinguir los detalles más pequeños de los objetos. Por supuesto una mayor resolución supone un mayor precio de la cámara y un tiempo de procesado de la imagen mayor, por lo tanto hay que sopesar la resolución con estos otros factores.

Relación calidad-ruido: Cuanto mejor sea esta relación, mayor será la calidad de la imagen capturada. Existen técnicas de filtrado, se verán más adelante, que sirven para eliminar el ruido eléctrico que se produce en el CCD. El ruido eléctrico produce como una ligera niebla en la imagen obtenida, que es necesario filtrar.

Programación vía RS232: Existen cámaras que permiten la calibración de todos sus parámetros, o de parte de ellos desde el ordenador directamente; lo que permite una automatización del proceso de calibrado.

Otras características: Existen cámaras con otras características como son **las cámaras progresivas de doble canal**, que permiten la adquisición en la mitad de tiempo ya que transfieren la imagen por dos canales a la vez, por uno las líneas y por otro lado las líneas impares, o los datos pares e impares, etc.;

2.1.1 CONCLUSIONES PARA LA SELECCIÓN DE LA CÁMARA

Según los puntos vistos anteriormente, se puede concluir que en la elección de la cámara, hay que:

Determinar que proceso se va a realizar y según éste decidir si es necesaria una cámara color o monocromática.

Elegir según el presupuesto, calidad necesaria, velocidad de adquisición, etc.; si la cámara debe ser analógica o digital.

Seleccionar dentro de las cámaras disponibles aquellas que tengan mejores características en su velocidad de obturación, características del CCD, tipos de sincronismos, programación vía software, etc. Buscar la cámara más completa dentro de nuestras posibilidades.

2.2 LA ÓPTICA

La óptica a utilizar dependerá especialmente de las **condiciones ambientales y de la distancia de medición**.

Dependiendo de la luminosidad ambiental, puede ser conveniente la utilización de filtros que la eliminen y que permitan capturar convenientemente los objetos a explorar. Por otro lado, se pueden usar filtros especiales para destacar un tipo de color determinado y poder realizar la segmentación de una forma más sencilla.

En la selección de la óptica, sobre todo cuando se van a determinar características geométricas de los objetos, **es necesario evitar todas aquellas lentes que deformen la imagen:** lentes que amplían o reducen el campo de visión, etc.

Se aconseja **estudiar las distancias y tamaño de los objetos a medir** para elegir la óptica más adecuada. Generalmente es interesante adquirir una lente provista de zoom y otra lente fija. Con el zoom se pueden realizar los ajustes necesarios y determinar que lente fija se tiene que comprar para la aplicación que se pretende realizar. Existen por otro lado sistemas de zoom automáticos controlados directamente desde la cámara.

Por último **es aconsejable proveerse de unos anillos de extensión** que den más posibilidades a la hora de aumentar o disminuir las distancias mediante el uso de lentes fijas. Estos anillos permiten tener un reducido número de lentes para casi cualquier distancia que sea necesaria.

En la Figura 2.1 se puede una tabla de distancias para diferentes ópticas, usando un juego de anillos de extensión formado por anillos de 0.5 mm, 1 mm, 5 mm, 10 mm, 20 mm y 40 mm de distancia.

Lens Focal Length		f = 12.5mm	f = 25mm	f = 50mm	f = 75mm	f = 8.5mm	f = 16mm
		inf. 0.3m	inf. 0.6m	inf. 1.0m	inf. 1.2m	inf. 0.2m	inf. 0.5m
EXTENSION TUBE KIT consist of 6 pcs. (0.5mm, 1.0mm, 5.0mm, 10mm, 20mm, 40mm)	0.5mm	36cm~20cm	129cm~43cm			19cm~12cm	56cm~28cm
	1.0mm	20cm~14cm	67cm~34cm			11cm~9cm	30cm~20cm
	1.5mm(1+0.5)	15cm~12cm	46cm~28cm	178cm~68cm		9cm~8cm	21cm~16cm
	5.0mm	8cm~8cm	17cm~15cm	60cm~41cm	129cm~68cm		9cm~8cm
	10mm		11cm~10cm	34cm~21cm	71cm~50cm		6cm~6cm
	15mm		9cm~8cm	26cm~23cm	51cm~41cm		
	20mm		8cm~8cm	22cm~20cm	42cm~35cm		
	25mm(20+5)			19cm~18cm	36cm~32cm		
	30mm(20+10)			17cm~17cm	32cm~29cm		
	35mm(20+10+5)			16cm~16cm	29cm~27cm		
	40mm			15cm~15cm	27cm~25cm		
	45mm(40+5)				25cm~24cm		
	50mm(40+10)				24cm~23cm		
	55mm(40+10+5)				23cm~22cm		
	60mm(40+20)				22cm~21cm		
65mm(40+20+5)				21cm~21cm			

Figura 2.1. Tabla de conversión de distancias para anillos de 0.5 mm, 1 mm, 5 mm, 10 mm, 20 mm y 40 mm.

2.2.1 CONCLUSIONES PARA LA SELECCIÓN DE LA ÓPTICA

De lo arriba descrito se puede llegar a las siguientes conclusiones:

Para el trabajo de ajuste es aconsejable adquirir una óptica de zoom que permita determinar la óptica fija a utilizar en cada caso práctico.

Es interesante plantearse el uso de filtros especiales para reducir la luz ambiental o para simplificar la imagen resultante.

Conseguir un juego de anillos permitirá tener una variedad mayor de distancias de enfoque.

2.3 LA TARJETA DE ADQUISICIÓN DE IMAGEN

Otro aspecto fundamental a considerar, dentro del SVA, es el de la tarjeta de captura de imagen. Existen multitud de tarjetas en el mercado orientadas hacia el terreno profesional, con gran cantidad de características que hace que sea difícil seleccionar unas u otras. A continuación, se van a exponer las más importantes:

Tipo de entrada de vídeo: Como ya se ha visto cuando se trataba de la forma de estudiar la elección de la cámara, la entrada de vídeo tiene que admitir la salida de vídeo de la cámara; sea vídeo compuesto, RGB o digital. Existen tarjetas en el mercado que permiten varios tipos de entradas. Por otro lado, puede ser conveniente la utilización de varias cámaras a la vez, por ejemplo en sistemas de visión tridimensional, siendo éste otro factor a tomar en cuenta para la elección de una tarjeta de visión.

Memoria en la tarjeta y controlador propio de DMA: Existen tarjetas en el mercado que usan la memoria del ordenador para almacenar las imágenes, mientras que otras disponen de gran cantidad de memoria propia. Las tarjetas que usan la memoria del ordenador suelen ser más baratas, pero suelen tener varios inconvenientes. Dos de los

inconvenientes más importantes que se advierte en este tipo de placas son: por un **lado el uso de la memoria del ordenador puede producir conflictos con el sistema operativo** por una mala gestión de la memoria, además el *driver* que se instala reduce la memoria que tiene el sistema operativo para otros programas; por otro lado estas tarjetas al tener que gestionar la memoria propia del sistema tienen que acceder al *chipset* del ordenador resultando muchas veces incompatibles con algunas arquitecturas que usan *chipsets* antiguos o diferentes a los especificados en el manual de la tarjeta de visión. La tarjeta que dispone de memoria en sí misma y además de un sistema de control DMA eficiente que permita llevarla hasta la memoria del ordenador es la selección más adecuada³. Realmente el uso o no de memoria en la tarjeta puede suponer el ocupar un 1% o 42% de tiempo de CPU según un tipo de tarjeta u otra⁴ (todo el tiempo que se gaste en el uso de la CPU en la transferencia de la imagen a memoria supone capacidad que perdemos para otros procesos).

Entradas y salidas de sincronismo: Es fundamental que la tarjeta de vídeo nos permita el control de las señales de sincronismo de la cámara, sobre todo cuando se va a realizar la adquisición de varias cámaras a la vez. Existen tarjetas que tienen una salida de sincronismo en vídeo compuesto que se suele utilizar para controlar una o más cámaras. Este proceso llamado *sync-locking* permite sincronizar señales cuando se realiza la conmutación entre canales o en el caso en el que se capturen imágenes de dos o tres cámaras a la vez en un solo paquete.

Posibilidad de uso de triggers externos, entradas y salidas digitales: Toda tarjeta de visión que se precie debe de tener entradas de *trigger* que permitan realizar un disparo cuando se produce un evento externo determinado. Supongamos un sistema de reconocimiento automático de etiquetas de botellas de vino, en el que la captura se realiza justo cuando un sensor detector de posición indica que hay una botella preparada para su reconocimiento. Existen muchas tarjetas con entradas y salidas digitales que permiten realizar un control sencillo sin necesitar una tarjeta adicional de adquisición digital.

Uso de DSP: Debido a que las aplicaciones que se realizan en un entorno industrial tienen que ser en tiempo real, es fundamental que la tarjeta disponga de un procesador digital de señal que pueda programarse para realizar tareas de preprocesado, reduciendo el tiempo de proceso que va a tener que realizar la CPU del *host*. Por supuesto, uno o varios DSPs con procesamiento en paralelo incrementan sustancialmente el precio final de la tarjeta de visión. Habrá que sopesar el coste frente al tiempo máximo de cálculo estimado.

Uso de procesadores especiales para proceso pipeline: Para la realización de **sistemas de visión en tiempo real**⁵ existen actualmente tarjetas formadas por procesadores especialmente diseñados para realizar el procesamiento en *pipeline* aumentando sustancialmente la velocidad de cálculo (ver Figura 2.2).

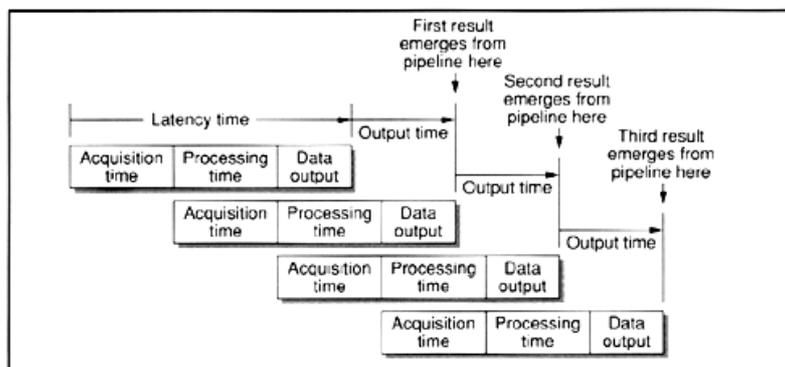


Figura 2.2. Proceso pipeline de visión en tiempo real.

Bus PCI vs Bus ISA-VESA: Actualmente los buses ISA/VESA han caído en desuso, por lo que se recomienda tarjetas para bus PCI ya que además tiene una capacidad de transferencia de hasta 132 MB/s en *burst mode*.

Software de adquisición y librerías: Una tarjeta por muy bien dotada que se encuentre, si no tiene unas librerías adecuadas que permitan sacarle jugo a sus características, no vale para nada. Este es un punto importante a tener en cuenta, que el software de adquisición y manejo de la tarjeta de visión se ajuste a nuestras necesidades.

3 Ver tutorial: *Why Machine Vision Applications Need Frame Grabbers with On-Board Memory*, Scott Israel, Imaging Technology Incorporated; en la dirección www.imaging.com de Internet.

4 Ver curso en Power Point con el mismo título que el punto anterior en la dirección www.imaging.com de Internet.

5 Ver tutorial: *Applying Real Time Machine Vision*, Ben Dawson, Imaging Technology Incorporated; en la dirección www.imaging.com de Internet.

2.3.1 CONCLUSIONES PARA LA SELECCIÓN DE LA TARJETA DE VISIÓN

Resumiendo, se puede llegar a las siguientes conclusiones:

Buscar una tarjeta que disponga de las entradas y salidas que se necesiten: formato de vídeo, número de cámaras que puede manejar, *triggers*, sincronismos, etc.

Elegir, siempre que se pueda, una tarjeta que disponga de memoria en su propia placa y un buen sistema de transferencia y control DMA.

Sopesar si va a ser necesario disponer de un DSP que trabaje en paralelo para acelerar las operaciones de procesado. Habrá que estudiar tanto las características del DSP como las herramientas software que permitan una programación rápida y eficaz del procesador digital de señal.

Si el proceso, tiene que realizar complejos cálculos y además tiene que implementarse en un sistema de tiempo real; será necesario seleccionar tarjetas más avanzadas donde se usen mecanismos *pipeline* en el procesado de la imagen.

No dejar de estudiar para cada tarjeta, dentro de las candidatas, el software de adquisición que lleva cada una; analizando los pros y los contras de cada uno de ellos.

2.4 EL HOST Y EL SOFTWARE UTILIZADO

Otro aspecto importante a considerar es el software utilizado y el ordenador que va a realizar todo el proceso de cálculo. Existen diferentes puntos a tener en cuenta: realmente hay que considerar que el ordenador elegido, el sistema operativo y el software utilizado están íntimamente relacionados:

2.4.1 SELECCIÓN DEL TIPO DE HOST

Claramente se puede llegar a la conclusión que un proceso de visión artificial requiere gran cantidad de cálculo y sobre todo si se quiere que el sistema actúe en tiempo real, una velocidad de procesado muy rápida. En el mercado existen máquinas de muy diversa índole diferenciándose especialmente por el procesador o procesadores que llevan. Esta diferencia es sustancial, porque permitirán un tipo de tarjetas u otras.

Dentro de la gama de sistemas que se pueden encontrar, exceptuando los sistemas propietarios que están actualmente desapareciendo, aparecen máquinas con las siguientes familias de procesadores: *INTEL (x86 y Pentium)*, *ALPHA*, *MIPS (Silicon Graphics, etc.)*, *PowerPC*, etc. Realmente hay que considerar que el ordenador va a trabajar en un ambiente industrial donde existen grandes campos magnéticos, polvo, humedad, cambios bruscos de calor, golpes, vibraciones, ruido y perturbaciones eléctricas, sobretensiones, etc.; esto lleva obligatoriamente a utilizar **ordenadores industriales**.

Los ordenadores industriales están actualmente muy preparados para el ambiente industrial teniendo carcasas que les protegen de todos los factores enumerados anteriormente. La mayoría de ordenadores industriales que hay en el mercado están basados en el uso de los procesadores de la familia INTEL encontrando placas de inserción compuestas por procesadores 80386 hasta placas de última generación con dos procesadores PENTIUM II a 450 MHz⁶.

Las ventajas de usar ordenadores industriales basados en la plataforma INTEL son numerosas: gran cantidad de fabricantes de todo tipo de periféricos, gran cantidad de información y software, sistema conocido por millones de personas, multitud de herramientas de programación a bajo precio, fácil mantenimiento, etc. Asimismo, existen gran variedad de fabricantes de tarjetas de captura de vídeo y de software de visión artificial que decantan rápidamente estas plataformas como las más idóneas.

Otra de las ventajas de las plataformas basada en procesadores INTEL o compatibles es el uso de las instrucciones MMX que sirven para mejorar sustancialmente el procesado de imagen.

⁶ El presente documento se escribe en Enero de 1999. Realmente la evolución de los procesadores PENTIUM es muy rápida, encontrándonos en pocos años con que estos procesadores se han quedado desfasados. Actualmente cuando sale un nuevo procesador, en pocos meses aparece una tarjeta para ordenador industrial que la implementa, se puede comprobar en cualquier revista de productos electrónicos como por ejemplo en "EPE Equipos Productos Electrónicos®" del ELSEVIER PRENSA.

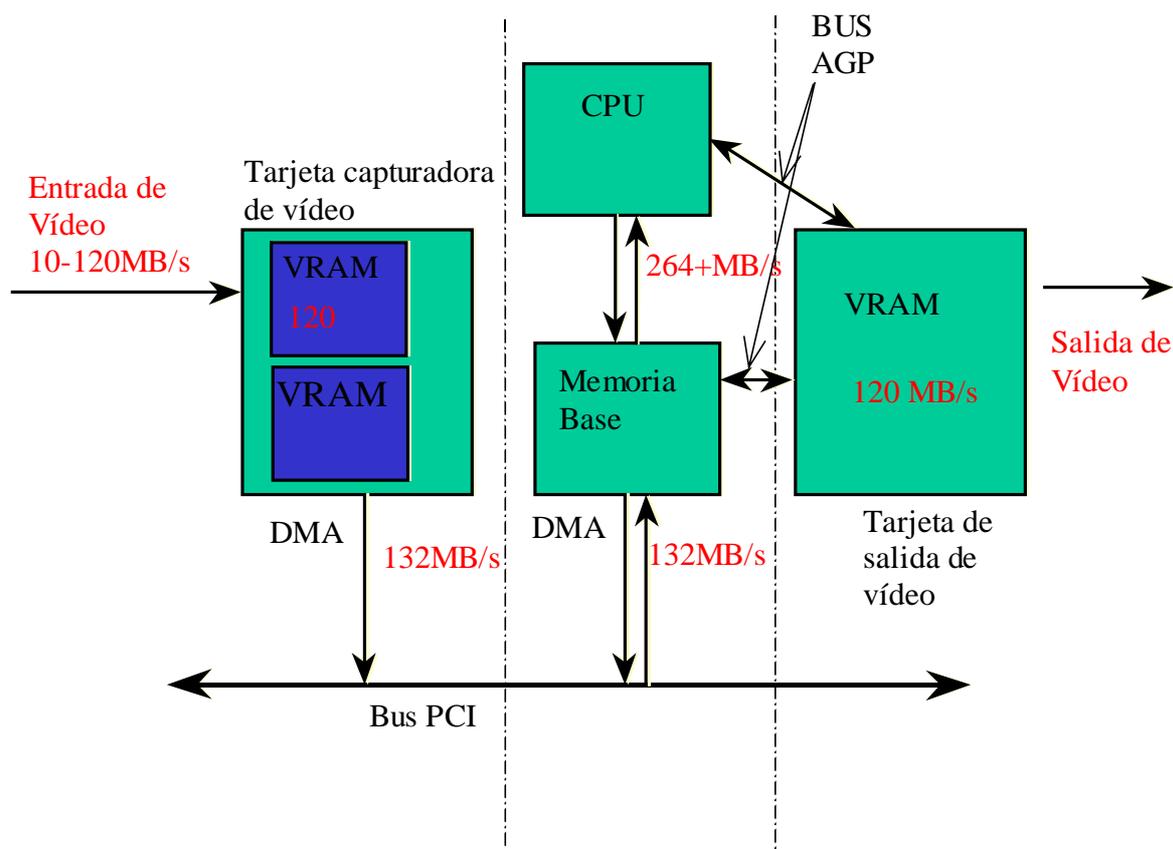


Figura 2.3. Sistema óptimo de captura de imagen mediante arquitectura INTEL y buses PCI y AGP.

Si el procesador es importante, también lo es el uso de una gran cantidad de memoria (no siendo problemático actualmente), de un buen disco duro, y de una tarjeta de visualización mediante BUS AGP que permita reducir los tiempos de transferencia en la visualización de las imágenes capturadas. El bus AGP permite a la tarjeta de salida de vídeo conectarse directamente a la CPU y la memoria base del sistema a 64 bits, acelerando sustancialmente todo el proceso de visualización. Es conveniente tener una buena tarjeta de salida de vídeo con suficiente cantidad de memoria.

Actualmente las tarjetas de vídeo han avanzado considerablemente, encontrándose en el mercado placas con procesadores digitales de señal especializados en transformaciones 2D y 3D, con precios muy competitivos. En la Figura 2.3 se puede observar el sistema *framegrabber* más adecuado para aplicaciones de visión artificial, está formado por una tarjeta de adquisición de imagen formada por dos bancos de VRAM donde la imagen es capturada con velocidades entre 10 y 100 Mbytes por segundo. Esta imagen captura es transferida en *burst mode* a la memoria del *host* con una velocidad de transferencia de 132 MB/s a 32 bits, evitándose así un cuello de botella. Por otro lado la tarjeta de visualización al usar el bus AGP a 64 bits, permite una velocidad de transferencia del doble 264MB/s. Todo este proceso se realiza mediante transferencia DMA, evitándose de esta manera ocupar ciclos del procesador que puede dedicarse a realizar otros procesos en paralelo.

2.4.2 ELECCIÓN DEL SOFTWARE DE PROGRAMACIÓN Y VISUALIZACIÓN

Tan importante como la elección del hardware es la del software a utilizar. Los programas de visión artificial profesionales están formados por las siguientes partes: entorno visual de pruebas, procesado y estudio; lenguajes de programación y librerías de desarrollo.

Generalmente los programas de visión artificial suelen tener un entorno donde se puede capturar la imagen, procesarla con diferentes algoritmos y determinar por ejemplo que funciones se van a utilizar. Además suelen tener un lenguaje de programación propio⁷, que permite la realización de subrutinas que no existan en sus propias librerías de desarrollo.

7 Por ejemplo el Vision Basic del Impuls Academic SDK XXL.

Por otro lado, es necesario estudiar las funciones que se ofrecen para el desarrollo de las aplicaciones reales. Estas funciones, realizadas la mayoría de las veces en C, C++ o ensamblador buscando la máxima velocidad posible; se suministran en librerías que se pueden utilizar con compiladores de C como: Microsoft Visual C++, Borland C++, Watcom C++ o Borland C++ Builder. Es necesario observar la cantidad de subrutinas⁸ de que constan las librerías para cada etapa del proceso de visión artificial: preprocesado (filtrados, ecualización por histogramas, transformaciones geométricas, etc.), segmentación (transformadas, erosión, dilatación, etc.), análisis, selección, etc.

Cuanto más completa sea la librería de funciones, más fácilmente se podrá solucionar los problemas que vayan surgiendo.

2.4.3 CONCLUSIONES PARA LA SELECCIÓN DEL ORDENADOR Y DEL SOFTWARE ADECUADO

De lo expuesto anteriormente se puede llegar a las siguientes conclusiones:

Las características más adecuadas del ordenador a utilizar serían: un ordenador industrial con uno o varios procesadores INTEL o compatibles lo más potentes posibles PENTIUM II a 450 o 400 MHz, AMD K6 3D Now, etc.; suficiente cantidad de memoria (32 o 64 MBytes de RAM mínimo) y un buen disco duro.

Un software lo más completo posible de visión artificial que permita por un lado la prueba de secuencias de algoritmos y que, por otro lado, tenga una librería de funciones lo más completa posible.

2.5 TÉCNICAS DE ILUMINACIÓN

Una vez elegidos los elementos constituyentes del SVA, se pasan a estudiar los elementos correspondientes al entorno y la forma de situarlos. **Un entorno debidamente controlado es imprescindible para obtener unas condiciones de partida óptimas** que aseguren una perfecta adquisición. Dentro del entorno aparecen como partes fundamentales que influyen fundamentalmente en las expectativas tanto de calidad como de la imagen buscada: la iluminación, el fondo, la posición de la cámara, etc.

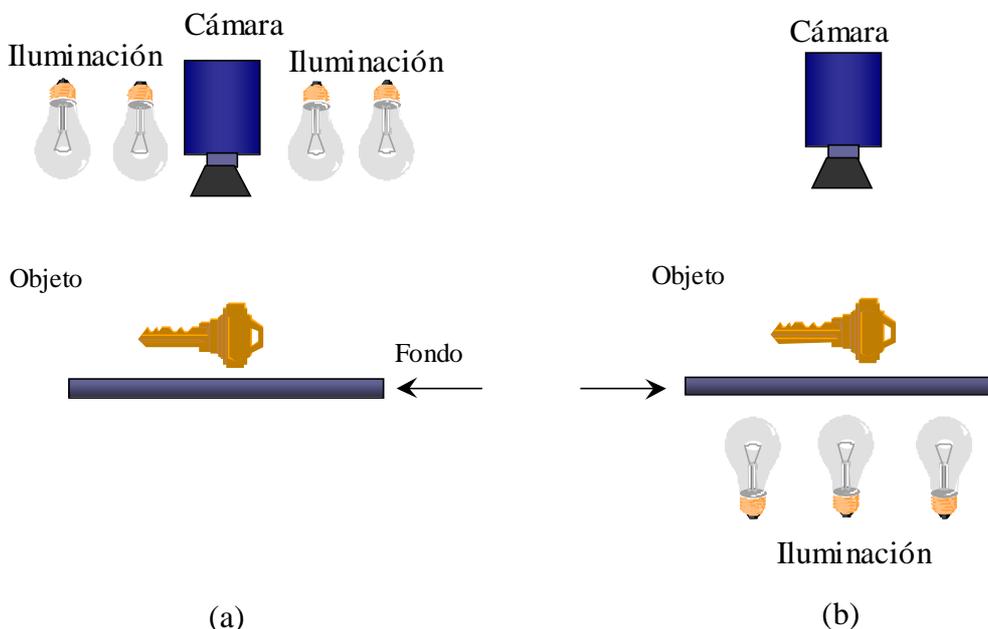


Figura 2.4. Tipos de iluminación: a) frontal y b) retroiluminación.

2.5.1 LA ILUMINACIÓN

La iluminación de la escena tiene que realizarse de una forma correcta, dada la importancia que tiene. Existen fundamentalmente dos formas de iluminación, como se puede observar en la Figura 2.4:

Iluminación frontal: Donde la luz incide directamente sobre el objeto, ya sea verticalmente, horizontalmente, de forma oblicua o de forma difusa.

Iluminación trasera o Retroiluminación: Donde se ilumina una pantalla de forma que lo que se busca es el contorno del objeto a modo de *sombra chinesca*. Este objeto puede estar delante o detrás de la pantalla.

La **iluminación frontal** permite distinguir los detalles de los objetos, así como su forma, permitiendo extraer más parámetros de cada objeto como su color, detalles internos, etc.; que permitan una mejor segmentación. Por otro lado, la **retroiluminación** sólo sirve para la detección de contornos simplificando por otra parte la etapa de segmentación que se debe efectuar posteriormente a la captura.

VENTAJAS Y DESVENTAJAS DE LA ILUMINACIÓN FRONTAL

Como ya se ha comentado en el párrafo anterior, la iluminación del objeto permite distinguir claramente los detalles del mismo, pero esta iluminación puede presentar diferentes inconvenientes.

La iluminación con puntos de luz fijos (focos de luz) va a producir dos fenómenos contraproducentes: **la creación de sombras y los reflejos**.

Las sombras que crean los focos de luz oblicuos, son un serio inconveniente para detectar correctamente las formas de los objetos, sobre todo cuando estos son oscuros. En términos generales el efecto de las sombras es algo que se trata de evitar.

Al usar focos de luz, en objetos brillantes se producen brillos que también son contraproducentes perjudicando seriamente la detección de los contornos y detalles de los objetos.



Figura 2.5. Iluminación con lámpara circular para un sistema de inspección

La mejor forma de evitar estos dos efectos, consiste en utilizar luz difusa ya sea con fuentes especiales que generan luz difusa (por ejemplo fibras ópticas que expanden la luz en todas direcciones), por el uso de lámparas circulares de luz que iluminen de forma homogénea (ver Figura 2.5) o por el uso de luz indirecta.

Por otro lado, como el sistema va a trabajar en un entorno industrial, es conveniente evitar las posibles interferencias externas (por ejemplo, el típico caso del carro de filoguiado que lleva encima una luz giratoria de forma que cada vez que pasa cerca del sistema de visión produce errores en la interpretación de la escena, siendo por otro lado difícil de detectar) mediante pantallas que reduzcan la incidencia de la luz de los focos externos.

VENTAJAS Y DESVENTAJAS DE LA RETROILUMINACIÓN

La iluminación por detrás del objeto puede hacerse de dos formas: situando el objeto entre la pantalla y los focos que lo iluminan, de forma que lo que captura la cámara es la sombra proyectada sobre la pantalla (Figura 2.6.a); o el objeto entre la cámara y la pantalla de forma que lo que se ilumina es el fondo de la escena (Figura 2.6.b).

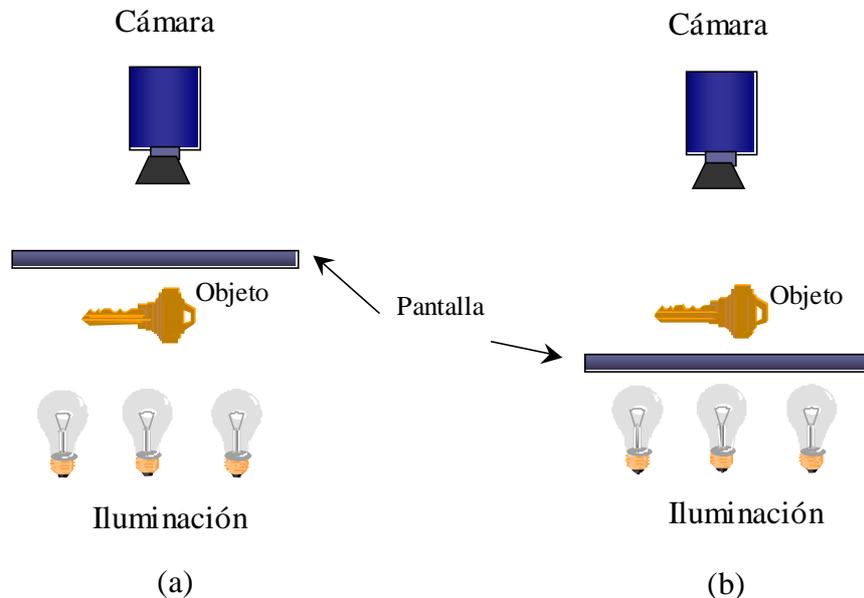


Figura 2.6. Retroiluminación: a) proyección sobre una pantalla, b) iluminación del fondo de la escena.

La proyección del objeto sobre la pantalla solo permite el reconocimiento del contorno de éste, teniendo como ventaja por otro lado **que el preprocesado y segmentación se realiza en pocos pasos**. Prácticamente con este sistema no es necesario realizar preprocesado y si los objetos están aislados unos de otros, la segmentación es sumamente sencilla. Por otro lado, si existen objetos “montados” uno encima de otro, el reconocimiento de ambos se torna muy difícil.

Un problema que aparece habitualmente cuando se proyecta un objeto en una pantalla es que si la iluminación no está completamente perpendicular con la pantalla, se deforma la sombra del objeto. Es conveniente, tomar precauciones para que esto no se produzca.

Por otro lado, el segundo método es más ventajoso. Se trata de iluminar el fondo de forma que sea más fácil diferenciarlo de los objetos. Este fondo suele ser una pantalla blanca que difumina la luz trasera produciendo un fuerte fondo blanco iluminado. Además, al ser más fuerte esta iluminación, se eliminan las sombras que puede producir la iluminación ambiental consiguiendo un fuerte contraste entre objeto y fondo. Debido a este contraste los detalles se diferencian mejor: pequeños agujeros, pequeñas deformaciones, etc.

Para que la segmentación sea lo más sencilla posible se busca diferenciar fácilmente el fondo de los objetos: ya sea por un color chillón o inexistente en el universo de objetos a reconocer, o con una iluminación blanca que sature al CCD dando niveles máximos de color (blanco nieve).

De todas formas la segmentación se complica si los objetos están conectados o unos encima de otros. Este método permite realizar un segundo nivel de segmentación ya que se captura cada objeto con sus detalles superficiales intrínsecos: color, formas, etc.

Una de las mayores desventajas de la retroiluminación es la forma de implementarla dentro de un sistema industrial automático. En la inspección de objetos que son trasladados con una cinta transportadora, si utilizamos este sistema, la iluminación tiene que estar debajo de la cinta transportadora. Es por lo tanto necesario el uso de una cinta translúcida, que actúe como pantalla, transportando los objetos a través del sistema de visión.

APLICACIONES DE LOS DIFERENTES SISTEMAS DE ILUMINACIÓN

La selección de la iluminación más adecuada para cada aplicación depende del tamaño, de las características de la superficie, de la geometría y de las necesidades del sistema. A continuación se analizan, de forma general, las aplicaciones de cada sistema de iluminación.

Sistemas anulares

Los iluminadores anulares o anillos de luz proporcionan una gran intensidad de luz uniforme, con ausencia de sombras, por lo que son adecuados para iluminar objetos de reducidas dimensiones.

Se pueden sujetar al objetivo de la cámara y se les puede acoplar filtros de colores, polarizadores / analizadores, y difusores para, así, eliminar reflejos y aumentar el efecto difusor.



Figura 2.7. Iluminador anular.

Sistemas de iluminación lineal

Estos sistemas son apropiados para iluminar intensamente zonas muy estrechas, siendo aplicables en entornos de visión industrial, metrología, procesos de inspección y, muy específicamente, con cámaras lineales en inspección en continuo de papel, tejidos, cerámica material metálico, madera, ...



Figura 2.8. Iluminador lineal

Sistemas de panel

Este tipo de iluminación se utiliza para formar una silueta de un objeto, depositándolo sobre la fuente de luz, cuando las estructuras superficiales del mismo no son importantes. Se utilizan tanto en visión industrial (metrología) como en microscopía, para hacer que el objeto aparezca oscuro sobre un fondo claro. Este tipo de iluminación también puede utilizarse para producir un efecto de luz difusa muy uniforme.

Entre las posibles aplicaciones se incluyen: inspección de vidrio, localización y medida de piezas, medida de productos opacos, detección de límites,... Estos sistemas de iluminación son también adecuados como fuente de luz homogénea proporcionando una luz uniformemente distribuida para la inspección de superficies.



Figura 2.9. Iluminador de panel

Sistemas puntuales

Estos sistemas están diseñados para iluminar intensamente desde distintas posiciones. Existen dos tipos de iluminadores puntuales tipo latiguillo: flexibles y semirrígidos. Estos últimos permiten orientar las terminaciones lumínicas sobre el objeto sin necesidad de sistemas de fijación específicos.

En las terminaciones se pueden conectar lentes con polarizadores para concentrar la luz y evitar brillos, y filtros de color y difusores para conseguir los efectos deseados para cada aplicación.

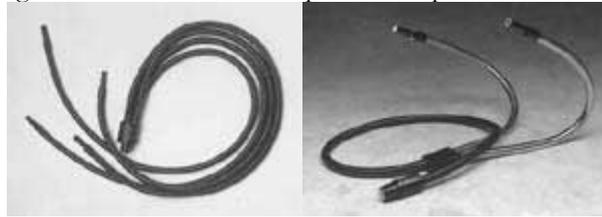


Figura 2.10. Iluminadores puntuales flexibles y semirrígidos.

Luz Difusa

Este tipo de iluminación se emplea en superficies brillantes y lisas donde cualquier reflexión puede confundirse con un defecto. Ejemplos de este tipo de aplicaciones incluyen la verificación de discos compactos y la inspección de patrones en circuitos impresos o placas electrónicas.

Láser

Los patrones láser se utilizan mayoritariamente en aplicaciones de medida de profundidad y de irregularidades de superficies, existiendo un gran número de formas y tamaños: una o múltiples líneas, línea punteada, matriz de puntos, circunferencia única o concéntricas, cruz, cuadro, rejilla, etc..

2.5.2 EL FONDO

El fondo de la escena cumple un papel esencial cuando se trata de simplificar alguna de las etapas subsiguientes (como puede ser la segmentación). Éste debe ser lo más homogéneo posible y de un color que permita distinguirlo fácilmente de los objetos. Cualquier mancha o defecto que existan en el fondo, puede ocasionar errores en la etapa de reconocimiento.

Utilizando iluminación frontal, el fondo debe de ser lo más opaco posible evitando todo reflejo. El color negro opaco suele ser el más utilizado.

Por otro lado, si la iluminación es trasera, se busca que la pantalla difumine lo más posible la luz de forma que se obtenga un fuerte fondo blanco que lo distinga fácilmente de los objetos.

2.5.3 CONCLUSIONES PARA LA SELECCIÓN DE LA ILUMINACIÓN Y DEL FONDO

De lo anteriormente expuesto se puede llegar a las siguientes conclusiones:

Hay que elegir la iluminación que mejor se ajuste a la aplicación evitando las sombras y reflejos.

Buscar que la iluminación sea difusa.

Proteger al sistema de toda iluminación espontánea del exterior que pueda interferir en el proceso. Se usarán paredes que reduzcan lo más posible la entrada de luz ambiental.

Elegir un fondo adecuado tanto en su color como en su homogeneidad.

2.6 CALIBRACIÓN

Para obtener una imagen correcta sin deformación, es necesario efectuar un perfecto calibrado⁹ de la posición de la cámara, tal y como se muestra en la Figura 2.11.

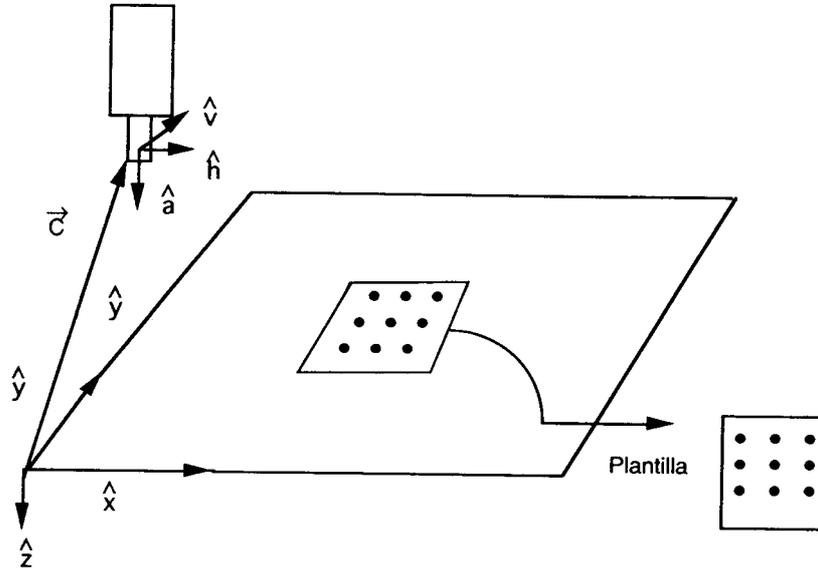


Figura 2.11. Calibrado de la posición de la cámara mediante una plantilla de 9 puntos.

Según la figura se pretende obtener los vectores unitarios \hat{a} , \hat{h} y \hat{v} ; y el vector de posición de la cámara \vec{C} . Para ello es preciso conocer la posición exacta de los nueve puntos de la plantilla, cuyas coordenadas serán:

$$\begin{aligned}\vec{P}_1 &= (p_{1x}, p_{1y}, p_{1z}) \\ \vec{P}_2 &= (p_{2x}, p_{2y}, p_{2z}) \quad (2.1) \\ &\dots\dots\dots \\ \vec{P}_9 &= (p_{9x}, p_{9y}, p_{9z})\end{aligned}$$

Tener estos puntos calibrados permite obtener los vectores unitarios y el vector de posición de la cámara. En realidad, el proceso de calibrado se simplifica para procesos en los que la cámara permanece perpendicular a la escena. Para ello, la plantilla se usa para conseguir que la perpendicularidad y la distancia sean correctas. Teniendo la plantilla con los nueve puntos y conociendo las dimensiones de estos y sus distancias de separación, el proceso de calibrado pasa por realizar un algoritmo que determine los centros de gravedad de los nueve puntos, sus tamaños y distancias de separación. De forma, que la cámara se ajuste para que todas las distancias entre puntos sean las mismas y el tamaño del cuadro sea el deseado.

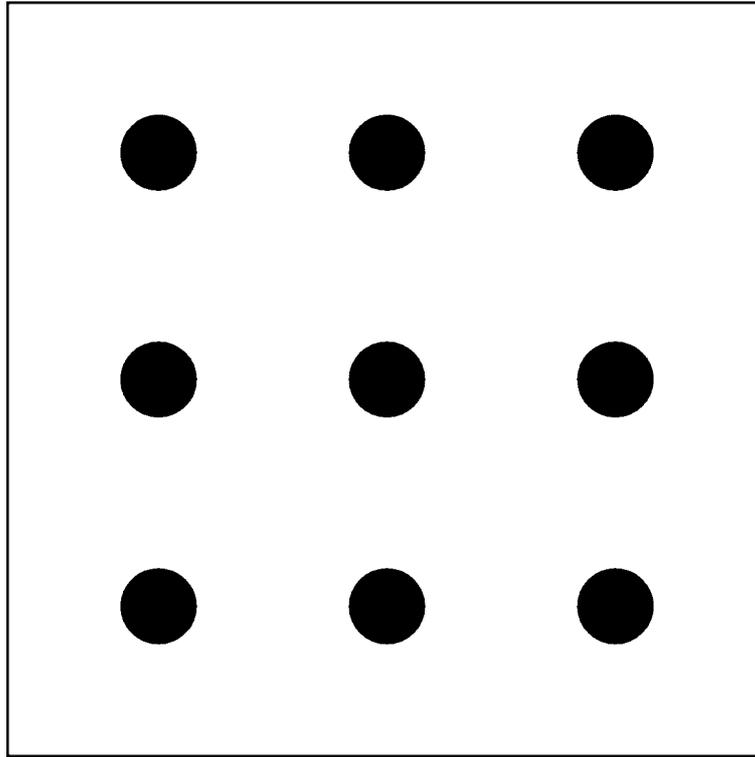


Figura 2.12. Plantilla de calibrado de puntos de 1cm diámetro, 3cm de distancia separación y 10 cm de lado.

2.6.1 CONCLUSIONES PARA REALIZAR UN CALIBRADO CORRECTO

De lo visto anteriormente, se puede llegar a las siguientes conclusiones:

Para realizar el calibrado lo mejor es utilizar una plantilla de nueve puntos donde se conocen sus dimensiones, distancias de separación, etc.

Será necesario realizar un programa de calibrado que determine de la imagen en tiempo real, los centros de gravedad de los puntos, la dimensión de cada punto, las distancias de separación, la redondez de cada punto, el lado horizontal y vertical del cuadro, etc.

Ajustar la cámara tanto en inclinación como en distancia para que exista la mayor igualdad posible entre todas las distancias y diámetros.

Una vez el sistema en funcionamiento, si se observan problemas en el reconocimiento, volver a realizar el calibrado. Hay veces que la vibración de las máquinas o un tornillo mal ajustado produce que la cámara se mueva.

3. PREPROCESADO

Toda imagen que se adquiere por medios ópticos, electroópticos o electrónicos sufre en cierta medida los efectos de la degradación que se manifiestan en forma de ruido, pérdida de definición y de fidelidad de la imagen. La **degradación** viene provocada por el ruido de los sensores de captura, imprecisiones en el enfoque de la cámara, movimiento de la misma o perturbaciones aleatorias, entre las que tiene relevancia el efecto de la propagación de la radiación en el medio de transmisión. Los mecanismos que tratan de contrarrestar estos efectos se incluyen dentro de la etapa de preprocesado, recibiendo el nombre de **operaciones de restauración**.

Generalmente **el preprocesado pretende reparar en la imagen los desperfectos producidos o no eliminados por el hardware**: deformación de ésta, ruido introducido, poco o mucho contraste o brillo, falta de equalización apropiada, etc. Los algoritmos de preprocesado permiten modificar la imagen para eliminar ruido, transformarla geoméricamente, mejorar la intensidad o el contraste, etc.; procesos que intentan mejorar el resultado final de la imagen capturada. Por supuesto, como estos algoritmos necesitan gran cantidad de tiempo de procesado, es lógico que lo mejor es utilizar un conveniente hardware que los evite.

En un proceso de visión artificial estos algoritmos tienen que ser utilizados lo menos posible, un uso excesivo de ellos repercutirá en el tiempo de proceso total e indicará que la calibración, iluminación y selección de los elementos de la etapa de adquisición no ha sido la adecuada.

A parte de la degradación de la imagen, existen características de la imagen que, en muchos casos, es conveniente mejorar. Así, muchas veces, se quiere mejorar el contraste, brillo, niveles de grises, eliminar brillos, aumentar los bordes, mejorar las texturas, etc. Todas estas técnicas vienen a denominarse **operaciones de mejora de la imagen**.

La parte de preprocesado en un sistema de visión artificial está formada por estos dos tipos de operaciones, es decir, todo el tratamiento de corrección y de mejora de la imagen que facilite el procesado de las siguientes etapas.

En este capítulo se expondrá el uso más práctico de las funciones comunes en preprocesado. De todas formas, existen algoritmos que son difíciles de clasificar en una etapa u otra, lo que produce que unos autores clasifiquen unos algoritmos en la etapa de preprocesado y otros en la de segmentación.

Tanto en la etapa de preprocesado, como en las de segmentación, extracción de parámetros y análisis de la información se ha realizado una serie de estudios de los mecanismos clásicos que existen en el procesado de imagen. Dentro de todos los algoritmos vistos, que utilizan estas técnicas, se han seleccionado aquellos que más utilidad puede tener en las aplicaciones de visión computacional. La descripción de estos algoritmos no pretende ser muy exhaustiva, si no que se basará en el uso práctico y aplicaciones para el reconocimiento de formas y la visión artificial (existiendo abundante documentación¹⁰ donde se muestra su funcionamiento y la forma de desarrollarlos en la práctica).

Para una mejor comprensión, en todo el tema de preprocesado y posteriores se utilizará como ejemplo una imagen de dos dimensiones $M \times N$ y de 256 niveles de grises por punto (siendo 0 negro carbón y 255 blanco nieve).

$$IM() \quad (3.1)$$

Cada punto se especificará por su posición i y j respecto de la esquina inferior izquierda:

$$IM(i,j) \quad (3.2)$$

Antes de comenzar con los algoritmos de preprocesado de imagen más comunes, se va a ver algunas operaciones elementales de procesado de imagen que se aplican en todas las etapas que se verán posteriormente.

3.1 OPERACIONES BÁSICAS

Existe gran cantidad de operaciones básicas utilizadas en visión artificial.

3.1.1 INVERSIÓN

Consiste en obtener el valor inverso ($1/\text{valor}$) del de cada pixel:

10 Ver Bibliografía.

$$IMB(i,j)=1/IMA(i,j) \quad (3.3)$$

No es muy utilizado, generalmente se usa para aspectos estéticos. **No hay que confundir la complementación con la inversión.**

3.1.2 OPERACIONES ARITMÉTICAS

ADICIÓN Y SUBTRACCIÓN

Consiste en sumar o restar imágenes unas con otras:

$$IMC(i,j)=IMA(i,j)\pm IMB(i,j) \quad (3.4)$$

Son muy utilizadas cuando se busca el movimiento de un objeto dentro de una escena. Se comparan dos imágenes correlativas en el tiempo mediante la substracción de una de ellas con la otra. Si algún objeto se ha movido se detecta inmediatamente porque aparecerá en la imagen resultante. Si la escena está completamente quieta, la imagen resultante será cero.

PRODUCTO O DIVISIÓN POR UNA CONSTANTE

Consiste en multiplicar o dividir el valor de cada pixel por una constante:

$$IMB(i,j)=C*IMA(i,j) \quad (3.5)$$

o

$$IMB(i,j)=IMA(i,j)/C \quad (3.6)$$

Con ello se aumenta o disminuye la luminancia de cada pixel, consiguiendo de esta forma **aumentar o disminuir el brillo** en una imagen.

LOGARITMO Y EXPONENCIAL

La función logaritmo:

$$IMB(i,j)=K*\text{Log}_{10}(1+IMA(i,j)) \quad (3.7)$$

se utiliza para aumentar el contraste de las zonas oscuras en detrimento de las zonas claras. Mientras que la función exponencial:

$$IMB(i,j)=K*\exp(IMA(i,j)-1) \quad (3.8)$$

se utiliza para aumentar el contraste de las zonas claras en detrimento de las oscuras.

3.1.3 OPERACIONES LÓGICAS

COMPLEMENTACIÓN (OBTENCIÓN DEL NEGATIVO)

Consiste en invertir el valor de los pixels de la imagen. Para la imagen ejemplo:

$$IMB(i,j)=255-IMA(i,j)=\sim IMA(i,j) \quad (\text{NOT}) \quad (3.9)$$

Se cambia el nivel de los grises de forma que los blancos nieve se convierten a negros carbón y los negros a blancos nieve. Es decir, el efecto que se obtiene es como el negativo de una foto.

AND, OR Y XOR

Considerando que los valores de cada pixel son *unsigned char*, tendríamos:

$$\text{IMC}(i,j)=\text{IMA}(i,j) \& \text{IMB}(i,j) \quad (\text{AND}) \quad (3.10)$$

$$\text{IMC}(i,j)=\text{IMA}(i,j) | \text{IMB}(i,j) \quad (\text{OR}) \quad (3.11)$$

$$\text{IMC}(i,j)=\text{IMA}(i,j) \wedge \text{IMB}(i,j) \quad (\text{XOR}) \quad (3.12)$$

para las funciones AND, OR y XOR respectivamente.

Son muy utilizadas en el uso de máscaras que sirven para obtener o eliminar una porción determinada de la imagen.

Está claro que, dentro de las operaciones lógicas, pueden aparecer múltiples combinaciones de estas.

3.1.4 TRANSFORMACIONES NO LINEALES

Funciones elementales tales como: corrección gamma, afilado de la imagen, etc.; se realizan mediante una función de transformación $h(x)$ (generalmente una tabla de 256 valores para 256 niveles de grises) de forma que la imagen resultante sería:

$$\text{IMB}(i,j)=h(\text{IMA}(i,j)) \quad (3.13)$$

3.1.5 SLICING

La traducción literal de *slice* implica porción, franja o sector. Se pretende el resalte de una determinada franja de niveles de gris que se dejan a un valor primario o máximo, mientras que los valores de luminosidad restantes se dejan a cero o a su valor previo, según se pretenda conservar o no el fondo.

En las Figura 3.1.c y Figura 3.1.d se pueden observar dos tipos de *slipping*, uno en el que los valores restantes se dejan a cero y el otro en el que se dejan a su valor original.

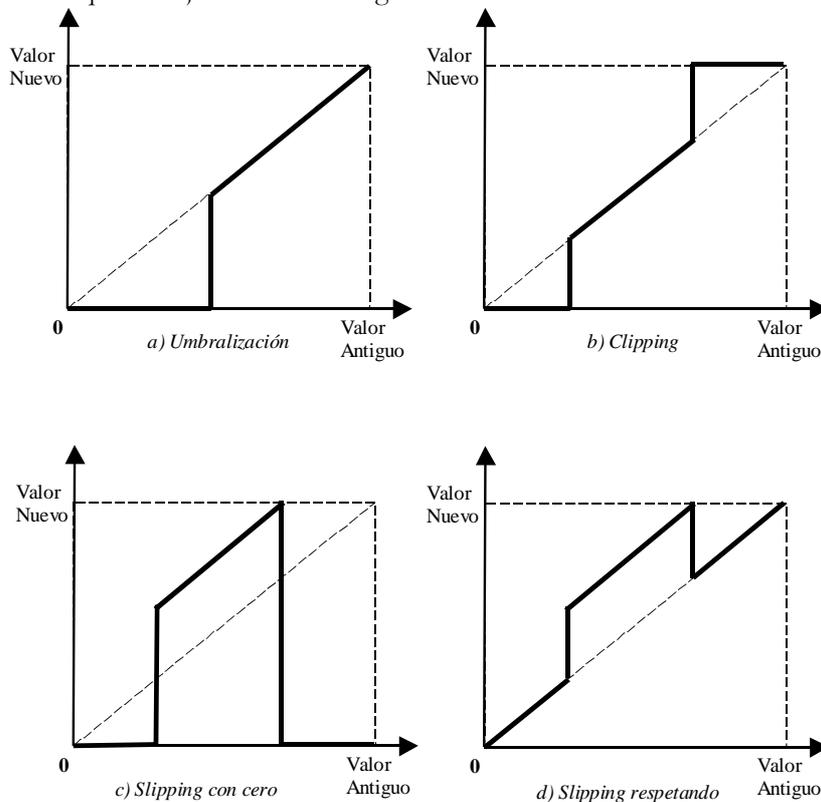


Figura 3.1. Valores resultantes en los niveles de luminancia de un pixel según el algoritmo utilizado.

3.1.6 CLIPPING

Consiste en convertir todos los niveles de grises superiores a un valor máximo val_{max} a un valor fijo k_{max} y los valores menores a un valor mínimo val_{min} a otro valor fijo k_{min} . Se puede resumir en:

$$IMB(i, j) = \begin{cases} K_{min} & \text{si } IMA(i, j) < val_{min} \\ IMA(i, j) & \text{si } val_{min} \leq IMA(i, j) \leq val_{max} \\ K_{max} & \text{si } IMA(i, j) > val_{max} \end{cases} \quad (3.14)$$

Esta función permite igualar todos los puntos que se consideran de un mismo origen, por ejemplo, con un fondo oscuro formado por pixels que tienen niveles de intensidad entre 0 y 20, es conveniente convertirlos todos a cero para simplificar posteriores análisis.

3.1.7 UMBRALIZACIÓN

También conocida como *thresholding* consiste en eliminar los valores superiores o inferiores (poniéndolos a cero) respecto a un valor conocido como **umbral**.

$$IMB(i,j)=IMA(i,j)*(IMA(i,j)\geq umbral) \quad (3.15)$$

3.1.8 BINARIZACIÓN

La binarización es una variante de la umbralización y consiste en dejar a cero todos los pixels menores de un umbral y a uno aquellos que son iguales o mayores, quedando constituida la imagen final por un conjunto de unos y ceros. El algoritmo es muy sencillo de implementar:

$$IMB(i,j)=(IMA(i,j)\geq umbral) \quad (3.16)$$

El resultado de la condición será 1 si es mayor o igual, o 0 si es menor que el valor de la constante *umbral*.

3.1.9 OTRAS FUNCIONES

Combinado las funciones vistas anteriormente entre ellas o con otros parámetros (valores de luminancia específicos, coordenadas de cada punto, valores de los puntos vecinos, etc.), se puede realizar gran cantidad de nuevos algoritmos con resultados muy variables. Enumerar todas las combinaciones sería prácticamente imposible, quedando como puerta abierta la experimentación práctica de algunas de ellas.

3.2 TRANSFORMACIONES GEOMÉTRICAS DE IMÁGENES

Estos algoritmos modifican las características geométricas de las imágenes, su uso fundamentalmente se aplica en la reconstrucción de imágenes deformadas, el giro y ajuste de las mismas o la deformación intencionada de ciertos rasgos para posteriores análisis. Generalmente los más utilizados son los de **escalado, translación, giro y espejo**.

Todos los algoritmos correspondientes a transformaciones geométricas se basan en realizar una nueva distribución de los pixels según lo que se pretenda. De esta forma el proceso de transformación geométrica se fundamenta en:

Determinar las nuevas coordenadas de cada *pixel* (i,j) en la rejilla transformada, (i',j') . Estas nuevas coordenadas (i',j') generalmente, no serán valores enteros. Este proceso depende del tipo de transformación a realizar.

Una vez obtenidos (i',j') hay que calcular los valores de los *pixels* (x,y) en la rejilla destino. Este proceso es común a todas las transformaciones y se denomina **interpolación**. La interpolación más utilizada es la **interpolación bilineal**¹¹.

El uso de las transformaciones geométricas en un entorno en tiempo real es muy reducido, debido al tiempo de proceso que necesitan. Se exponen las aplicaciones más comunes de cada tipo de algoritmo.

11 Alberto Domingo Ajenjo. *Tratamiento digital de imágenes*. Editorial ANAYA MULTIMEDIA.

3.2.1 ALGORITMOS DE TRASLACIÓN O DESPLAZAMIENTO

Consiste en sustituir cada pixel por el correspondiente a sus coordenadas más el desplazamiento en cada dirección k y l .

Si los desplazamientos son enteros para ambas direcciones el proceso consiste en sustituir cada *pixel* (i,j) por su *pixel* $(i+k, j+l)$, mientras que si tienen parte decimal es necesario realizar una interpolación.

Resumiendo, la función de translación sería:

$$IMB(i,j)=IMA(i+k,j+l) \quad (3.17)$$

En las aplicaciones prácticas el proceso de translación o desplazamiento se utiliza cuando se quiere posicionar cierto objeto detectado en un punto determinado para realizarle posteriores procesos, como por ejemplo: el uso de funciones lógicas AND, OR, XOR, etc.; con máscaras para obtener o eliminar ciertas partes de éste, procesos de unión o fusión con otras imágenes, etc.

3.2.2 ALGORITMOS DE ROTACIÓN O GIRO

Los algoritmos de giro son generalmente los más complejos y por lo tanto los más costosos en tiempo de procesado. Debido a esto, sólo se utilizan cuando es posible obtener una posición de giro que simplifique más posteriores procesos.

Dado un punto $IMB(i,j)$ y se rota θ grados, las coordenadas i' y j' del nuevo punto serán:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\text{sen } \theta \\ \text{sen } \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} \quad (3.18)$$

Por lo tanto, la versión girada de la imagen principal será:

$$IMB(i',j')=IMB(i \cos \theta - j \text{sen } \theta, i \text{sen } \theta + j \cos \theta)=IMA(i,j) \quad (3.19)$$

Después será necesaria realizar una interpolación.

Por ejemplo, si se tiene una imagen formada por múltiples rectángulos y cuadrados situados de forma perpendicular entre ellos y la imagen está girada, es muy útil alinearlos respecto a los ejes x e y de la imagen¹². De esta forma los procesos posteriores de segmentación y análisis sobre estos rectángulos o líneas serán más rápidos y eficaces.

3.2.3 ALGORITMOS DE ESCALADO Y DE ZOOM

Los algoritmos de escalado permiten reducir o aumentar la imagen, así como realizar el *zoom* de ciertas partes de la imagen.

Dado un factor de escalado α para las coordenadas x y otro β para las coordenadas y , obtendríamos:

$$IMB(k,l)=IMB(\alpha \cdot i, \beta \cdot j)=IMA(i,j) \quad (3.20)$$

Requiriendo una interpolación cuando i/α o j/β no son enteros.

El uso de las funciones de escalado no se circunscribe únicamente a la etapa de preprocesado sino que pueden incluirse en pasos de otras etapas. Por ejemplo, existen procesos posteriores (erosionados, reducción de la resolución, etc.) que no necesitan tanta cantidad de información, reduciéndose la imagen para que esos procesos sean más rápidos ya que tratan con una imagen de tamaño reducido.

3.2.4 ESPEJOS

Algoritmos que realizan un espejo respecto de la horizontal, vertical o diagonal.

Si la imagen es de dimensiones $M \times N$:

$$IMB(i,j)=IMA(N-i,j) \quad (\text{Espejo vertical}) \quad (3.21)$$

$$IMB(i,j)=IMA(i,M-j) \quad (\text{Espejo horizontal}) \quad (3.22)$$

¹² Ver ejemplo de uso de la transformada Radon en el manual de la Toolbox de Imagen de MatLab v5.2, donde se gira la imagen de un circuito integrado para situarlo alineado respecto de los ejes x e y de la pantalla (páginas 6-23 a 6-25).

$$\text{IMB}(i,j)=\text{IMA}(N-i,M-j) \text{ (Espejo diagonal) } (3.23)$$

Espejos se pueden realizar de múltiples formas, dependiendo de los ejes elegidos para realizarlo.

3.2.5 ALGORITMOS QUE REDUCEN LA RESOLUCIÓN DE LA IMAGEN

Actualmente, existen técnicas que utilizan redes neuronales para el reconocimiento de objetos mediante el análisis de varias resoluciones de una misma imagen¹³. Estas técnicas se basan en reconocer un objeto previo comparando la imagen de este objeto en diferentes resoluciones a la vez, para ellos se usan redes neuronales que relacionan unas con otras.

Estos algoritmos obtienen de una imagen de $M \times N$ pixels otras del mismo tamaño, donde se han reducido sus resoluciones, a un $1/2$, $1/4$, etc.

El procedimiento es muy sencillo, por ejemplo para reducir la resolución a la mitad consiste en coger cuatro pixels adyacentes, calcular su media e igualarlos todos a este resultado.

3.2.6 DEFORMACIÓN

La deformación de un objeto se puede realizar de múltiples formas. Se va a explicar una deformación sencilla en el eje x , siendo el mismo algoritmo para el eje y , simplemente cambiando las variables.

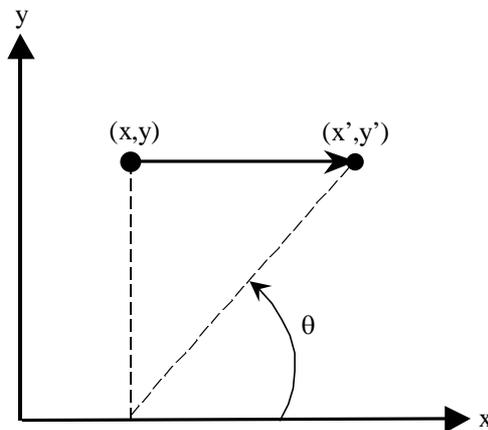


Figura 3.2. Deformación de un objeto en el eje x .

La ecuación de deformación en el eje x , tal como se ve en la Figura 3.2, viene dada por:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} 1 & \cot \theta \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} (3.24)$$

La transformación por lo tanto sería:

$$\text{IMB}(i',j')=\text{IMB}(i+j \cdot \cot \theta,j)=\text{IMA}(i,j) (3.25)$$

Teniéndose que truncar la coordenada $i+j \cdot \cot \theta$.

3.2.7 OTRAS TRANSFORMACIONES GEOMÉTRICAS

De la combinación de las anteriores transformaciones, puede surgir otro tipo de transformaciones geométricas: fundido, hinchado, *morphing*, etc. La mayoría de estos algoritmos son muy utilizados en los programas comerciales de diseño digital, existiendo abundante bibliografía sobre el tema que puede ampliar la lista de transformaciones geométricas aquí descritas.

13 Susan S.Young, Peter D.Scott, Cesar Bandera. *Foveal Automatic Target Recognition Using a Multiresolution Neural Network*. IEEE TRANSACTIONS ON IMAGE PROCESSING. VOL. 7, N° 8. AGOSTO 1998

3.3 ALGORITMOS GENERADORES DE RUIDO

Los algoritmos generadores de ruido¹⁴ permiten introducir ruido en una imagen para, por ejemplo, probar la respuesta del sistema de visión a diferentes interferencias que se puedan producir.

Antes de entrar a explicar el funcionamiento de estos algoritmos, es conveniente entender los diferentes tipos de ruido que pueden introducirse en una imagen, esto permitirá comprender en cada caso qué filtro habrá que utilizar así como el generador de ruido necesario para probarlo.

3.3.1 GENERADORES DE RUIDO DEPENDIENTE DE LA SEÑAL

Ruido dependiente de la señal: Este tipo de ruido viene producido generalmente en los propios sensores fotoeléctricos (CCD por ejemplo) o también debido al grano de la película de una fotografía o cinta de vídeo. Es un ruido dependiente de la señal. La función generadora de este tipo de ruido viene determinada por el producto de la imagen original con una función ruido. Esta función ruido intenta simular el ruido dependiente de la señal mediante una función normal-logarítmica y el ruido térmico, que se produce en los sensores fotoeléctricos, mediante la suma de una función de ruido Gaussiano. Resumiendo en la literatura especializada se define la siguiente transformación:

$$g(i,j) = c_2 \cdot (h(i,j) ** f(i,j))^{\gamma} + (c_2 \cdot (h(i,j) ** f(i,j))^{\gamma})^{1/2} \cdot n(i,j) + n_t(i,j) \quad (3.26)$$

siendo $f(i,j)$ la imagen original, $n(i,j)$ el ruido dependiente de la señal, $n_t(i,j)$ el ruido térmico y $b(i,j)$ la función de transferencia del sistema óptico. El símbolo ** denota una convolución de dos dimensiones.

3.3.2 GENERADORES DE RUIDO ALEATORIO

Ruido *salt-pepper* o ruido impulsivo: Está formado por puntos blancos o/y negros que se distribuyen de manera aleatoria por la imagen, este tipo de ruido es independiente depende de la señal. Es un ruido que aparece muchas veces producido por interferencias atmosféricas, o por acciones hechas por el humano (motores de coches, etc.). Se puede modelar de la siguiente manera:

$$g(i,j) = \begin{cases} z(i,j) & \text{con probabilidad } p \\ f(i,j) & \text{con probabilidad } 1 - p \end{cases} \quad (3.27)$$

siendo $z(i,j)$ un valor fijo (0 o 255) o variable, y $f(i,j)$ la imagen original que va a recibir el ruido.

Existen otras técnicas para realizar diversos tipos de ruidos: para generar ruido blanco multiplicativo gaussiano, ruido Laplaciano, etc.; que pueden ser consultados en la bibliografía descrita al final de este documento.

3.4 FILTRADO

Tal como se ha visto en el apartado anterior, existen diversos tipos de ruidos que se producen por múltiples circunstancias: los sensores, ruidos eléctricos, perturbaciones en el medio de transmisión, efectos térmicos, campos electromagnéticos, etc.

La mayoría de las implementaciones de filtros se realizan en dos dominios: **el dominio espacial** y **el dominio frecuencial**.

Los métodos basados en el dominio espacial hacen referencia a la manipulación directa de la luminancia de los *pixels*, mientras que los métodos basados en el dominio frecuencial modifican indirectamente la luminancia de cada *pixel* utilizando como factores de ponderación los valores de los otros *pixels* de la imagen o del entorno del punto y las relaciones numéricas entre ellos.

Las operaciones espaciales se pueden entender como la modificación directa del espectro de frecuencias de la imagen, por supuesto, este espectro de frecuencias se definen como frecuencias espaciales para una imagen. El proceso consistiría en obtener el espectro de la imagen y del filtro a aplicar, y multiplicar ambos para realizar el filtrado.

$$G(u,v) = F(u,v) \cdot H(u,v) \quad (3.28)$$

siendo $H(u,v)$ la transformada de Fourier del filtro en cuestión, y $F(u,v)$ y $G(u,v)$ las transformadas de las imágenes original y filtrada respectivamente. Debido a que las transformaciones de la imagen original y del filtro, y la transformada inversa requieren gran cantidad de tiempo de cómputo, es necesario recurrir a herramientas matemáticas que las mejoren.

Los filtros se pueden dividir en:

Filtros simples basados en la conceptualización de los resultados que produce un filtro: normalizaciones, filtros de media, dilataciones y algunos filtros paso-banda.

Filtros basados en el uso de la convolución. Como es bien sabido la convolución es el equivalente en el dominio del tiempo al producto de dos funciones en el dominio de la frecuencia.

Filtros basados en histogramas.

3.4.1 FILTROS FRECUENCIALES

Las técnicas basadas en filtros de dominio frecuencial se basan en la Transformada Discreta de Fourier (DFT) y la Transformada Discreta de Fourier Inversa (IDFT)¹⁵. Ambas se han desarrollado como un caso particular de enorme interés práctico de la *Transformada de Fourier* clásica o continua a partir de la aparición de la tecnología informática, que no trabaja con señales continuas.

TRANSFORMADA DISCRETA DE FOURIER (DFT)

Mediante esta transformada se obtiene el espectro frecuencial de la imagen en cada eje, obteniendo a unas frecuencias determinadas el modulo y argumento de cada componente frecuencial de la imagen. Quedaría:

$$E(u) = \frac{1}{N} \sum_{k=0}^{N-1} e(k) \ell^{-j \frac{2\pi}{N} ku} \quad (3.29)$$

donde $e(k)$ es la señal discreta en el dominio espacial; $u = 0,1,2,\dots, N-1$, es la frecuencia de la señal transformada y $E(u)$ es la señal discreta en el dominio frecuencial.

Es decir, se ha pasado de una señal discreta en el dominio espacial, $e(k)$, a una señal también discreta pero en dominio frecuencial, $E(u)$.

TRANSFORMADA DISCRETA DE FOURIER INVERSA (IDFT)

Mediante esta transformada se obtiene una imagen del formato de la imagen original, a partir del espectro frecuencial de esa misma imagen. Quedaría:

$$e(k) = \sum_{u=0}^{N-1} E(u) \cdot \ell^{j \frac{2\pi}{N} ku} \quad (3.30)$$

Es decir, se ha pasado una imagen del dominio frecuencial al dominio espacial, operación inversa a la anteriormente explicada.

El problema de estas transformaciones es que requieren gran cantidad de tiempo de cómputo por lo que es necesario recurrir a herramientas matemáticas que las mejoren, como la FFT (*Fast Fourier Transform*).

TRANSFORMADA RÁPIDA DE FOURIER (FFT)

La transformada rápida de Fourier (FFT) es una herramienta muy potente para el análisis frecuencial de señales discretas. La FFT permite convertir una señal que está en el dominio del tiempo al dominio de las frecuencias.

La FFT es un *ingenio* matemático que permite obtener de una señal digital, muestreada a lo largo del tiempo, los armónicos que la componen, en amplitud y desfase. Por ejemplo, si se tiene una FFT de 256 muestras y se realiza

15 Darío Maravall Gómez-Allende. *Reconocimiento de formas y visión artificial*. RA-MA. Ioannis Pitas. *Digital Image Processing Algorithms*. PRENTICE HALL.

la FFT a una señal muestreada a 10kHz, se obtendrán 256 números complejos formados por 128 válidos ya que los otros 128 son resultados espejos de los primeros. Estos 128 complejos de salida determinan la amplitud y desfase de 128 armónicos correspondientes a la señal muestreada. Es decir, los armónicos que componen la señal original. Para obtener la frecuencia, amplitud y desfase de cada armónico se realizan los siguientes cálculos:

Frecuencia de cada armónico: Para el valor complejo de salida número N de la FFT de longitud 256, la frecuencia correspondiente se obtiene de la siguiente manera:

$$W(N) = N \cdot \frac{\text{Frecuencia de adquisición} / 2}{\text{Longitud FFT} / 2} = \frac{10.000}{256} \quad (3.31)$$

Amplitud del cada armónico: Si $a_N + b_N i$ es el valor del complejo de salida de la función FFT para el armónico N será:

$$\text{Amp}(N) = \sqrt{a^2 + b^2} \quad (3.32)$$

Desfase de cada armónico: Obviamente el ángulo de desfase de ese armónico será:

$$\text{Desfase}(N) = \arctg \frac{b}{a} \quad (3.33)$$

De esta forma se obtiene el análisis frecuencial de una señal digital muestreada a lo largo del tiempo.

Realmente una imagen puede considerarse como una señal digital muestreada a lo largo del tiempo, con la salvedad de que en este caso esta señal es bidimensional y en vez de ser muestreada a lo largo del tiempo, es muestreada a lo largo de los ejes x e y . Debido a esto, la FFT tiene que ser bidimensional, existiendo algoritmos desarrollados para tal caso.

Existe toda una literatura especializada en el análisis digital de señal donde se estudian múltiples algoritmos de FFT, tanto unidimensionales como bidimensionales buscando la mayor eficiencia en velocidad posible.

Con la FFT se obtiene la transformada frecuencial $F(s)$ de la imagen. Si ésta se multiplica por la transformada $H(s)$ de un filtro determinado (paso-bajo, paso-alto, paso-banda o rechazo-banda) eliminando aquellos armónicos que no interesan, se consigue realizar el filtrado de la misma. Sólo quedará realizar la transformada inversa IFFT (también hay una amplia bibliografía al respecto) del producto de $F(s) H(s)$ para obtener la imagen filtrada. Por supuesto tanto la FFT como la IFFT tienen que ser bidimensionales.

La FFT no sólo se utiliza para el filtrado de señales, sino que tiene múltiples aplicaciones que más adelante se comentarán: determinar la forma, orientación, etc., de objetos; búsqueda del enfoque óptimo, eliminación de ruido periódico, etc..

Además de la FFT existen otros algoritmos matemáticos que también son muy usados en el tratamiento digital de imagen. Por ejemplo, la transformada discreta del coseno es muy utilizada para la compresión de imágenes apareciendo en formatos JPEG (formato de imágenes estáticas) y MPEG (formato de imágenes en movimiento).

FILTROS FIR Y FILTROS IIR

El filtrado digital de señal mediante filtros FIR (filtros de respuesta impulsional finita) y filtros IIR (filtros de respuesta impulsional infinita) también se pueden aplicar a imágenes. La forma de hacerlo consiste en aplicar la convolución (equivalente al producto en el dominio de la frecuencia) de una matriz bidimensional, formada por los coeficientes del filtro, con la matriz de la imagen a filtrar.

A continuación, se verán diferentes filtros que consisten en la convolución bidimensional de una matriz de coeficientes con los valores de luminancia de los puntos que rodean al *pixel* que se desea calcular.

Por ejemplo, un filtro constituido por una matriz:

$$h = \frac{1}{9} \cdot \begin{bmatrix} C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 \\ C_7 & C_8 & C_9 \end{bmatrix} \quad (3.34)$$

se calculará para el pixel $IMA(i,j)$ de una imagen, de la siguiente forma:

$$IMB(i, j) = \frac{1}{9} \cdot (C_1 \cdot IMA(i-1, j-1) + C_2 \cdot IMA(i, j-1) + \dots + C_8 \cdot IMA(i, j+1) + C_9 \cdot IMA(i+1, j+1)) \quad (3.35)$$

Realmente los filtros IIR son poco utilizados debido a que son bastante inestables. Las características más importantes de unos y de otros son:

Los filtros FIR:

Siempre son estables.

Son de fase lineal.

Mientras que los IIR:

Pueden ser inestables.

Nunca son de fase lineal.

Para resumir sobre filtrado frecuencial, es correcto decir que, en el sentido estricto de imágenes digitales, tiene poca aplicación en el contexto de los sistemas de visión artificial por dos razones:

En primer lugar por la carga computacional que conllevan las DFT (aunque se utilice la versión FFT), carga que hace inviable su aplicación en numerosas situaciones prácticas, sobre todo en sistemas en tiempo real.

En segundo lugar porque se pueden conseguir los mismos objetivos sin recurrir al filtrado frecuencial.

Por el contrario desde el punto de vista didáctico y de formación a la comprensión de las imágenes digitales, el filtrado frecuencial es muy interesante.

3.4.2 FILTROS ESPACIALES

Este tipo de filtros suele consistir en recorrer toda la imagen, pixel a pixel, y realizar alguna operación aritmética con un número concreto de pixels vecinos. Al conjunto de estos vecinos se les denomina ventana, la cual podrá ser de 3x3, 5x5, ...

En la Figura 3.3 se puede ver la representación de una ventana de 3x3 y de 5x5, en la que los pixels vecinos van del 1 al 9 y del 1 al 25 respectivamente, siendo el pixel central sobre el que se aplica un algoritmo. La mayoría de los algoritmos suelen ser combinaciones lineales de los pixels de la ventana. Para representar un filtro se suele dar el vector o matriz de convolución o núcleo que representa la combinación lineal entre ellos.

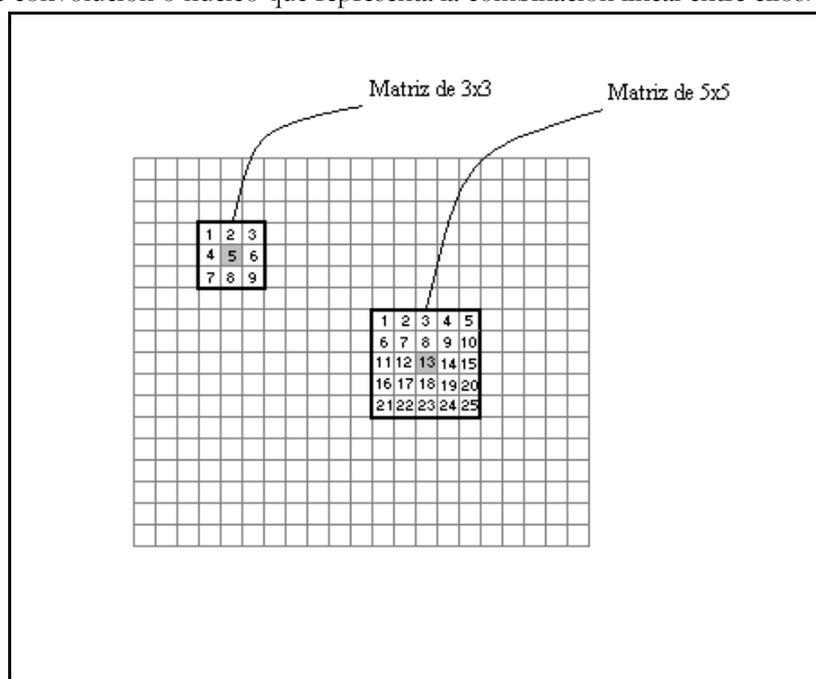


Figura 3.3. Representación de dos tipos de ventana para filtrado de imágenes.

PROMEDIADORES DE ÁREA

Son filtros paso-bajo que se utilizan fundamentalmente para reducir el ruido de alta frecuencia que se produce en una imagen. Para utilizarlos hay que jugar con el tamaño de la matriz y el número de veces que se realiza el filtrado. Si el tamaño o el número de pasadas es elevado el efecto del filtro será mayor pero se los bordes quedarán muy difuminados.

Existen múltiples matrices de convolución. A continuación se muestran las más comunes:

$$\begin{aligned}
 h_a &= \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} & h_b &= \frac{1}{10} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} & h_c &= \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \\
 h_d &= \frac{1}{25} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} & h_e &= \frac{1}{120} \cdot \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 7 & 13 & 7 & 1 \\ 2 & 13 & 24 & 13 & 2 \\ 1 & 7 & 13 & 7 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}
 \end{aligned} \tag{3.36}$$

Dentro de estas matrices de convolución hay que destacar h_c y h_e que producen un filtro promediador de área con menor rizado que los demás.

FILTRO DE MEDIA O DE SUAVIZADO

Consiste en sustituir cada pixel por la media aritmética de los puntos que tiene alrededor (incluido él mismo). Se toma una ventana de NxN puntos, donde el punto a sustituir es el central, se obtiene la media de la suma de todos los valores de los pixels de la ventana y se sustituye el pixel en la nueva imagen por el valor obtenido. El efecto del filtro aumenta a medida que lo hace el tamaño de la ventana.

Por ejemplo, para realizar un filtro con ventana de N=5 (es conveniente que sea impar para que el punto central de la ventana sea el que se quiere modificar), el algoritmo de filtro de media para cada *pixel* será:

$$IMB(i, j) = \frac{\sum_{m=-\frac{N}{2}}^{\frac{N}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} IMA(i+m, j+n)}{N \cdot N} \tag{3.37}$$

Los resultados que producen son equivalentes a una **difuminación** de la imagen produciendo un suavizado de los gradientes en toda la imagen. Es de gran utilidad para eliminar ruido espúreos producidos por un muestreo deficiente o una transmisión ruidosa.

Realmente el filtro de media es equivalente a un filtro promediador de área como el que usa la matriz h_a de la ecuación 3.34.

FILTRO DE MEDIANA

El filtro de mediana es muy parecido al filtro anterior. Consiste en obtener la lista de todos los valores de los pixels de la ventana, ordenarlos y coger el del medio (*mediana*), es decir, el valor que tenga igual número de valores superiores e inferiores a él dentro de la ventana.

Este filtro, al igual que el anterior, es muy usado para eliminar el ruido impulsivo de una imagen. La ventaja de éste con el anterior, es que en el filtro de mediana los detalles abruptos y bordes de los objetos (correspondientes a las partes altas de frecuencias) no se difuminan tanto.

FILTRO DE MODA

La *moda* de un conjunto de valores se define como el valor que más se repite dentro de ellos. Por lo tanto, el filtro de moda consiste en calcular el valor más repetido dentro de todos los *pixels* de una ventana.

Los efectos que se producen, dependen en gran manera del tipo de imagen que se filtre. Si ésta está formada por texturas que tiene gran cantidad de puntos dispares, el resultado será bastante significativo y variable, ya que en cada punto los valores serán muy dispares. Si, en cambio, la imagen tiene zonas homogéneas, este filtro las homogeneizará cada vez más.

NORMALIZACIÓN

Normalizar consiste en aplicar un mapa de transformación de luminancias para extender estas cuando la adquisición de la imagen se ha hecho con un rango menor. Por ejemplo, si se captura una imagen con 128 niveles de grises y se tiene la posibilidad de utilizar 256, el proceso de normalización consistiría en ampliar el rango de 128 a 256 niveles.

Aunque puede parecer que es una operación al *pixel*, hay que darse cuenta que para obtener los rangos máximo y mínimo, primero hay que contabilizar todos los puntos para determinar el mínimo valor y el máximo.

Se considera que K_{min} y K_{max} son los rangos mínimo y máximo de la imagen, y que el rango máximo que se puede visualizar es K . La transformación resultante sería:

$$IMB(i, j) = K \cdot \frac{IMA(i, j) - K_{min}}{K_{max} - K_{min}} \quad (3.38)$$

FILTROS PROMEDIADORES DE LÍNEA

Estos filtros realizan un promediado de la imagen pero sólo de líneas, con lo que sólo realizan el filtrado en la dirección en la que promedian. Este tipo de filtros respetan las altas frecuencias y, por lo tanto, los bordes en la dirección perpendicular a la que promedian. Pueden utilizarse cuando se quiere filtrar la imagen pero se quiere respetar los bordes de los objetos en su dirección perpendicular.

Se muestran algunas de las matrices de promediado, que pueden utilizarse tanto para el eje x como para el eje y .

$$\begin{aligned} h_f &= \frac{1}{5} \cdot [1 \ 1 \ 1 \ 1 \ 1] & h_g &= \frac{1}{9} \cdot [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\ h_h &= \frac{1}{111} \cdot [2 \ 5 \ 14 \ 22 \ 25 \ 22 \ 14 \ 5 \ 2] \end{aligned} \quad (3.39)$$

FILTROS REALZADORES DE CONTORNO

Son utilizados para hacer que los bordes sean más marcados. Realmente acentúan los puntos que más se diferencian de los vecinos, aumentando por lo tanto las altas frecuencias. En este caso el ruido de alta frecuencia aumenta.

Las matrices más utilizadas son:

$$h_i = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad h_j = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad h_k = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad (3.40)$$

FILTROS DE BÚSQUEDA DE GRADIENTE

Este tipo de filtros buscan y acentúan aquellos puntos donde el gradiente es mayor. El gradiente viene definido por:

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \quad (3.41)$$

dando esta función como resultado un vector cuya dirección indica la dirección de máxima diferencia entre puntos (la pendiente) y cuyo módulo indica el valor de la pendiente en esa dirección.

Para las direcciones horizontal, vertical y diagonales, las matrices son:

$$h_{GH1} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad h_{GH2} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{ horizontales }) \quad (3.42)$$

$$h_{GV1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad h_{GV2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad (\text{ verticales }) \quad (3.43)$$

$$h_{GDA1} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad h_{GDA2} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{ diagonales 1 }) \quad (3.44)$$

$$h_{GDB1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad h_{GDB2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (\text{ diagonales 2 }) \quad (3.45)$$

Dependiendo de la dirección del gradiente que se quiera acentuar, se utilizarán unas matrices u otras. Si se quiere acentuar el gradiente que se produce en varias direcciones, se calculará para cada punto el gradiente en cada dirección y después se tomará el valor medio, máximo o mínimo de los obtenidos.

Los gradientes que se obtienen con las matrices de la primera columna generan un desplazamiento de medio pixel en la imagen, pero son más fieles al gradiente ideal. En cambio los gradientes de la segunda columna no desplazan la imagen pero no se aproximan tanto al gradiente ideal.

FILTROS DETECTORES DE BORDES MEDIANTE GRADIENTES CON PROMEDIADO

Estos operadores obtienen el gradiente en una dirección y promedian en la perpendicular. Por ejemplo, el operador *Sobel* horizontal calcula el gradiente horizontal pero promediando el vertical.

Se pueden observar las matrices de convolución para tres de los métodos más usados de extracción de bordes: *Kirsch*, *Prewitt* y *Sobel*.

<i>Kirsh</i>	<i>Prewitt</i>	<i>Sobel</i>
$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

$$\begin{array}{ccc}
 \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & -2 \end{bmatrix} \\
 \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \\
 \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} & \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \\
 \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\
 \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \\
 \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \\
 \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}
 \end{array} \tag{3.46}$$

donde se obtienen los bordes para ocho direcciones: N, NO, O, SO, S, SE, E y NE. Si se quiere detectar un borde en una dirección determinada, hay que utilizar sólo una de las matrices. En cambio, si se quiere detectar el borde en todas las dimensiones, habrá que hacer para cada *pixel* ocho convoluciones usando en cada una la matriz de una dirección y después obtener el valor medio, el máximo o el mínimo, según se pretenda que los efectos sean más o menos acentuados.

En otras bibliografías aparecen otras máscaras asignadas para los detectores *Prewitt* y *Kirsh*, quedando a merced del lector el uso de unas u otras según sus resultados.

FILTROS DETECTORES DE BORDES CON EL OPERADOR LAPLACIANO

El operador laplaciano se define como:

$$L[f(x, y)] = \partial^2 \frac{f(x, y)}{dx^2} + \delta^2 \frac{f(x, y)}{dy^2} \tag{3.47}$$

desarrollando¹⁶ para una matriz de 3x3 puntos, se obtiene:

$$L_{IMB(i,j)} = IMA(i, j-1) + IMA(i-1, j) + IMA(i, j+1) + IMA(i+1, j+1) - 4 \cdot IMA(i, j) \tag{3.48}$$

16 Ver literatura especializada.

que traducido a una matriz de convolución queda:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (3.49)$$

La ventaja del laplaciano frente a los gradientes de primer orden, es que detecta mejor los bordes cuando las variaciones de intensidad de la imagen no son lo suficientemente abruptas. El problema es que estos suelen ser muy sensibles al ruido, lo que unido a la incapacidad de detectar la dirección del borde, los hace muy limitados para el tratamiento digital de la imagen. De todas formas, si se quiere utilizar el filtro laplaciano, es conveniente atenuar la sensibilidad al ruido utilizando filtros ponderados que incluyan un efecto de filtrado paso bajo.

Generalmente, para obtener un buen filtro, se realiza la transformada inversa de Fourier del perfil de filtro deseado y así, se determina la máscara bidimensional.

FILTROS DETECTORES DE PUNTOS Y DE LÍNEAS

Se puede detectar fácilmente los **puntos aislados** con la aplicación de la máscara siguiente:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3.50)$$

donde se advierte claramente el significado de esta matriz de convolución.

Para detectar **líneas**, se utilizan las siguientes matrices:

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad (\text{horizontales}) \quad (3.51)$$

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad (\text{verticales}) \quad (3.52)$$

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad (\text{diagonal 1}) \quad (3.53)$$

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad (\text{diagonal 2}) \quad (3.54)$$

FILTROS DETECTORES DE BORDES: EL OPERADOR DE HOMOGENEIDAD

Consiste en calcular el máximo de la diferencia del punto central con todos los que le rodean. Tomando, por ejemplo, un área de puntos con los siguientes valores:

$$\begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix} \quad (3.55)$$

se calcula el máximo de la diferencia de la luminancia del punto central con los que tiene alrededor: $\max \{|5-1|, |5-2|, |5-3|, |5-4|, |5-6|, |5-7|, |5-8|, |5-9|\}=4$, y se sustituye el *pixel* por el nuevo valor. Para obtener un buen resultado, es importante determinar un **umbral** (ajustable experimentalmente) que convierta el resultado en un valor binario.

FILTROS DETECTORES DE BORDES: EL OPERADOR DIFERENCIA

Consiste en obtener el máximo de la diferencia entre puntos opuestos. Para la matriz 3.53, el algoritmo sería el siguiente: $\max \{|1-9|, |7-3|, |4-6|, |2-8|\}=8$; es decir, obtener el máximo de la resta absoluta de la luminancia del punto de la esquina superior izquierda con el de la inferior derecha, de la luminancia del punto de la esquina superior derecha con el de la esquina inferior izquierda, etc.

Este tipo de filtro también necesita un umbral para que los resultados sean satisfactorios.

Las ventajas de este filtro y el anterior son su rapidez de proceso, ya que no hay que realizar productos y se puede implementar fácilmente por hardware.

FILTROS DETECTORES DE BORDES: DIFERENCIAS DE GAUSSIANOS O EL SOMBRERO MEJICANO

Las máscaras de convolución 3.54 y 3.55 son muy utilizadas cuando se quiere detectar aquellos bordes de los objetos grandes y no los de los objetos pequeños. El centro de las máscaras es de un valor positivo y se curva hacia un valor negativo terminando en cero, la forma de la segunda máscara se aproxima bastante a la de un sombrero mejicano.

$$\begin{bmatrix} 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \\ -1 & -3 & 5 & 16 & 5 & -3 & -1 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 \end{bmatrix} \quad (3.56)$$

$$\begin{bmatrix} 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & -2 & -3 & -3 & -3 & -3 & -3 & -2 & 0 \\ 0 & -3 & -2 & -1 & -1 & -1 & -2 & -3 & 0 \\ -1 & -3 & -1 & 9 & 9 & 9 & -1 & -3 & -1 \\ -1 & -3 & -1 & 9 & 19 & 9 & -1 & -3 & -1 \\ -1 & -3 & -1 & 9 & 9 & 9 & -1 & -3 & -1 \\ 0 & -3 & -2 & -1 & -1 & -1 & -2 & -3 & 0 \\ 0 & -2 & -3 & -3 & -3 & -3 & -3 & -2 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix} \quad (3.57)$$

OTROS FILTROS DETECTORES DE BORDES

Existen otros tipos de filtros detectores de bordes que son muy usados debido a sus buenos efectos en la práctica.

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (3.58)$$

3.4.3 OTROS ALGORITMOS DE FILTRADO

Aunque se han descrito numerosas técnicas de filtrado, existen muchas más que han aparecido últimamente o que van apareciendo día a día. Se describen brevemente algunas de ellas:

El filtro *Wiener*¹⁷ se utiliza para mejorar imágenes borrosas aplicando una transformación de la imagen y su inversa para obtener la función de transferencia del filtro. Lo que se busca es minimizar una función de error del cuadrado de la resta de una señal bidimensional aleatoria con una imagen estimada formada por la imagen con ruido convolucionada con un filtro lineal.

Método de interpolación *spline* que trata de ajustar una curva a los *pixels* observados y deducir de ella el ruido a partir de su segunda derivada.

Filtros adaptativos que usan métodos estadísticos para actuar en diferentes zonas de la imagen de forma distinta. Existe una combinación muy usada, donde se usa el filtro Wiener adaptándose a cada parte de la imagen, de esta forma se realiza un filtrado preservándose los bordes y otras partes de alta frecuencia de las imágenes.

Como se ve, las técnicas de filtrado son muy numerosas y existe un campo muy amplio de estudio de la aplicación de éstas al tratamiento de imagen.

Realmente, en las técnicas de visión artificial, se busca la eficiencia en resultados y tiempo de cálculo, ya que los procesos generalmente tienen que ser en tiempo real, y eso significa que no se puede perder mucho tiempo en el filtrado de la imagen. Por esto, en el campo de la visión computacional se suelen utilizar los algoritmos más comunes y rápidos de implementar y ejecutar.

3.5 OPERACIONES BASADAS EN HISTOGRAMA

El **histograma** de una imagen es una curva donde se representa la frecuencia con que aparece cada nivel de gris en la imagen. Generalmente se suele representar en el eje *x* el nivel de gris (por ejemplo de 0 a 255) y en el eje *y* el porcentaje de cada nivel de gris.

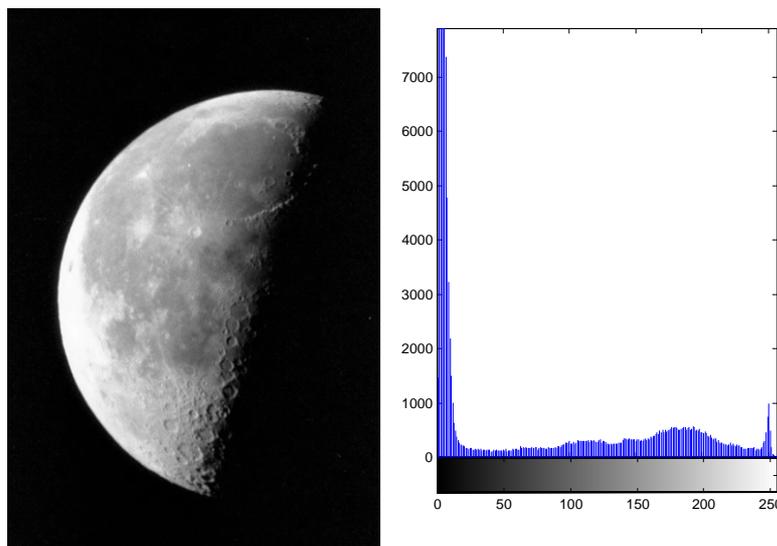


Figura 3.4. Imagen y su histograma donde se puede observar la gran cantidad de puntos oscuros.

Observando el histograma, se puede obtener una idea aproximada de la distribución de niveles de grises de la imagen: dónde está la mayor parte de la información para, por ejemplo, poder realizar una compresión de la imagen; se puede utilizar para la segmentación, etc.

3.5.1 TRANSFORMACIONES DEL HISTOGRAMA

Una vez obtenido el histograma de una imagen, se puede multiplicar por una nueva función de transferencia que permite modificar el contraste de la imagen de una forma muy sencilla. El algoritmo se encarga de sustituir cada nivel de luminancia de los *pixels* de la imagen por un nuevo valor obtenido de una tabla de transformación.

FUNCIÓN CUADRADO

La primera transformación que se va a ver, es la función cuadrado:

$$f_c(v) = v^2 \quad \text{para } v = 0..1 \quad (3.59)$$

Cuya función de transferencia será:

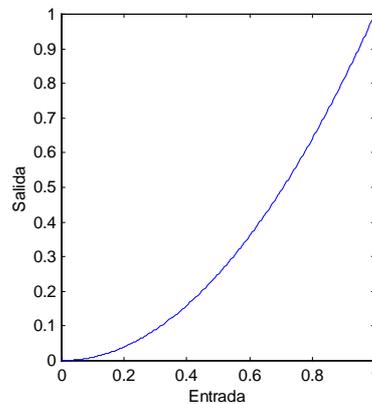


Figura 3.5. Función de transferencia de la transformación cuadrática.

De forma que cada pixel se transformará, de la siguiente forma:

$$IMB(i, j) = abs \left[f_c \left(\frac{IMA(i, j)}{255} \right) \cdot 255 \right] \quad (3.60)$$

El resultado de esta transformación es un oscurecimiento general, mejorando el contraste de los niveles bajos de gris (es decir los puntos oscuros) pero empeorando el contraste de los niveles altos.

FUNCIÓN CÚBICA

Su función de transferencia es:

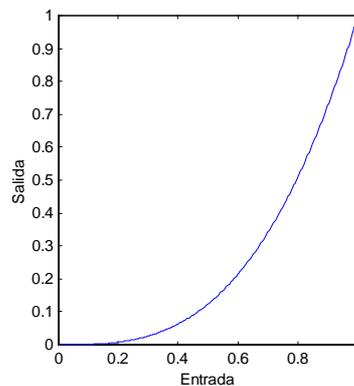


Figura 3.6. Función de transferencia de la transformación cúbica.

Que se obtiene de la siguiente forma:

$$f_c(v) = v^3 \quad \text{para } v = 0..1 \quad (3.61)$$

Utilizando la ecuación 3.58 con esta nueva función de transferencia se obtiene un resultado mucho más acentuado que la función cuadrado.

FUNCIÓN RAÍZ CUADRADA

Aplicando:

$$f_c(v) = v^{\frac{1}{2}} \quad \text{para } v = 0..1 \quad (3.62)$$

Se obtiene la siguiente función de transferencia:

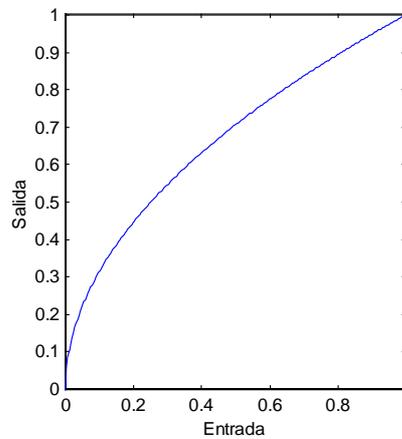


Figura 3.7. Función de transferencia de la transformación raíz cuadrada.

Cuyos resultados son contrarios a los obtenidos con la función cuadrado.

FUNCIÓN RAÍZ CÚBICA

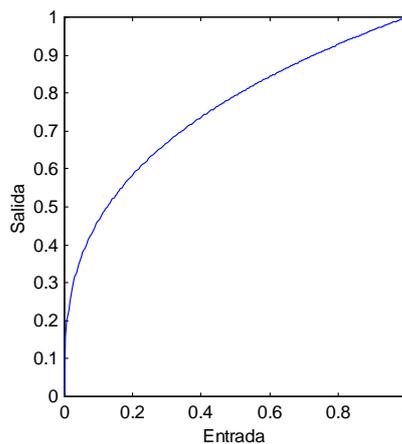


Figura 3.8. Función de transferencia de la transformación raíz cúbica.

Como se ha visto en anteriores transformaciones:

$$f_c(v) = v^{\frac{1}{3}} \quad \text{para } v = 0..1 \quad (3.63)$$

En la Figura 3.8, se puede observar la función de transferencia que genera. Ésta acentúa más los efectos que produce en el blanqueo de la imagen comparándola con la función raíz cuadrada.

OTRAS TRANSFORMACIONES

Como se ha visto, se pueden realizar gran cantidad de transformaciones dependiendo de la forma de la curva de la función de transferencia. Ésta se puede realizar de forma experimental ajustándola para obtener los resultados deseados.

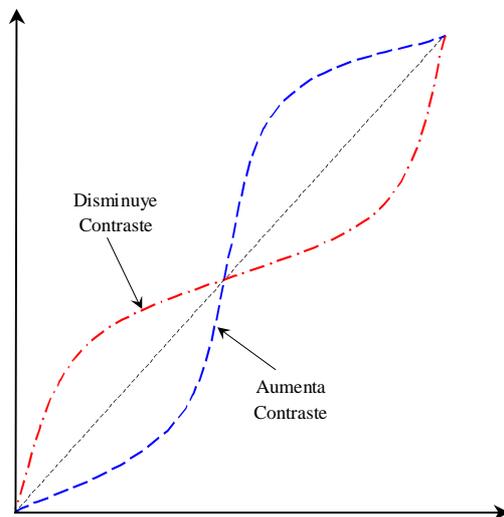


Figura 3.9. Función de transferencia para aumentar y disminuir el contraste de las zonas claras y oscuras.

Existen dos funciones de transferencia interesantes que permiten aumentar o reducir el contraste de las partes altas y bajas de los niveles de luminancia, combinando la función cuadrática para una zona y la función raíz cuadrada para la otra, o viceversa.

Además de este tipo de transformaciones, existen otras muy conocidas que utilizan curvas para aumentar y reducir el contraste, como por ejemplo el **factor de corrección gamma** que aparece en múltiples programas de diseño y tratamiento de la imagen. Realmente este factor si es igual a uno es equivalente a la línea diagonal de la Figura 3.9. Si éste es menor de 1 la curva pasa por encima de la recta diagonal, y si es mayor la curva pasa por debajo.

3.5.2 ECUALIZACIÓN DEL HISTOGRAMA

Cuando una imagen presenta dentro de su histograma una gran cantidad de puntos en torno a un nivel de gris central, es muy probable que sea una imagen con un contraste pobre. Todo esto es debido a que los márgenes de las luminosidades se traducen a un rango muy pequeño de niveles de grises, y por lo tanto, se desaprovecha todo el rango de valores de luminancia disponibles.

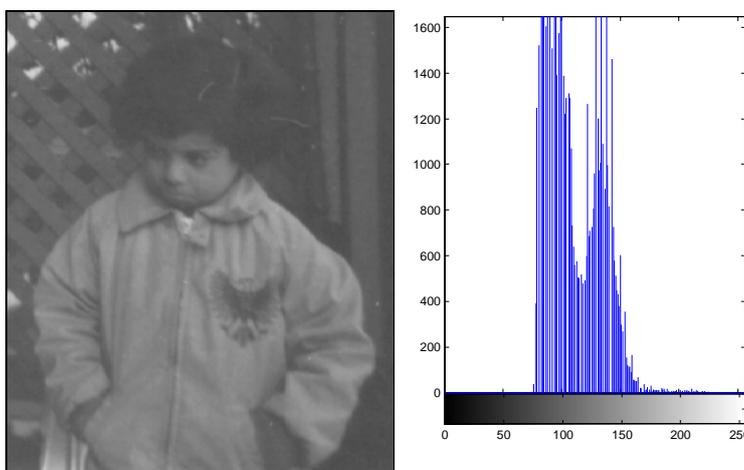


Figura 3.10. Imagen original y su histograma.

La ecualización del histograma trata de repartir los *pixels*, de forma que la luminancia de éstos esté más distribuida, consiguiendo, de esta forma, aumentar el contraste y distinguir mejor los distintos objetos que forman la imagen. Como defecto fundamental, la ecualización del histograma tiende a aumentar el ruido.

Para exponer la forma de realizar la ecualización de un histograma, es conveniente traducir éste a una función de densidad de probabilidad $f_d p$ de una variable r que varía de 0 a 1. Esta variable corresponde a los niveles de gris debidamente normalizados (en este caso divididos por 255 para 256 niveles de grises). La respuesta de la función $f_d p$ es simplemente el valor del histograma para cada nivel de gris normalizado entre 0 y 1.

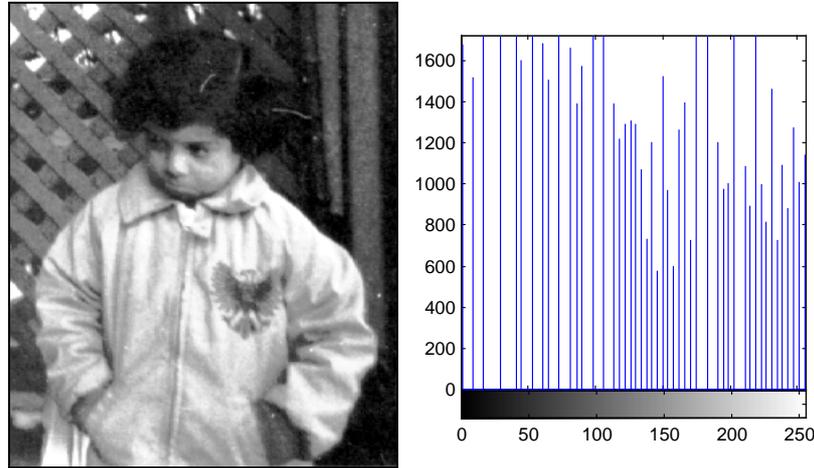


Figura 3.11. Imagen ecualizada y su histograma.

La operación de ecualización del histograma, consiste en obtener uno nuevo con una distribución uniforme de los *pixels* en los diferentes niveles de gris. La nueva función de transferencia $s=T(r)$ produce un histograma plano cumpliendo la siguiente relación:

$$f_s(s) = f_r(r) \cdot \left. \frac{dr}{ds} \right|_{r=T^{-1}(s)} \quad (3.64)$$

Para poder conseguir que $f_s(s)=1$ es preciso aplicar la transformación acumulativa¹⁸, que para funciones discontinuas como las de un histograma real, se aproxima a:

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} \quad (3.65)$$

donde la $f_d p$ asociada al histograma es:

$$f_r(r_k) = \frac{n_k}{n} \quad 0 \leq r_k \leq 1 \quad (3.66)$$

$$k = 0, 1, \dots, 2^q - 1$$

siendo q el número de bits del digitalizador, n_k el número de *pixels* de la imagen digital con el nivel de intensidad normalizado r_k y n el número de *pixels* totales.

4. SEGMENTACIÓN

Segmentar una imagen digital significa dividirla en zonas disjuntas e individualizadas. Es decir, consiste en diferenciar los diversos objetos y dónde se encuentran del fondo, que puede ser más o menos complejo, de la imagen.

Al final de la etapa de segmentación, se tienen que conocer perfectamente los objetos que hay para extraer las características propias de cada uno de ellos. Además, cada *pixel* de la imagen tiene que tener una etiqueta que los defina, de forma que simplemente por agrupación de puntos con la misma etiqueta y conectados espacialmente, se pueda determinar la lista de objetos (estos objetos son realmente zonas o regiones individualizadas dentro de la imagen, ya que un objeto, en el sentido estricto de la palabra, puede estar repartido en varias regiones diferentes dentro de la imagen obtenida).

Realmente, la etapa de segmentación es crucial para el reconocimiento de formas pudiéndose complicar o simplificar enormemente según sea la escena más o menos compleja. En un entorno industrial puede ocurrir que las escenas se compliquen más o menos, dependiendo del conjunto de objetos que se tengan que reconocer y de la disposición de estos en el entorno. Dentro de las técnicas de segmentación que se van a analizar, se van a tratar fundamentalmente aquellas técnicas que se aplican para disposiciones de objetos no muy complejas, donde: **Los objetos no estén “montados” unos sobre otros.** Esto producirá que la medida del contorno de los objetos se complique enormemente.

Los objetos tengan unos niveles de intensidad uniformes o cuasiuniformes, que permitan distinguirlos del fondo fácilmente.

Estas técnicas también se pueden utilizar en aplicaciones industriales donde los objetos se montan unos sobre otros, siempre que no sea necesario obtener el contorno de los objetos y éste sea conocido (por ejemplo el uso de un sistema de visión para contar el número de piezas iguales, aunque éstas estén unas sobre otras).

Por otro lado, se supondrá que los objetos tienen unos niveles de luminancia uniformes que los distinguen fácilmente del fondo en el que están situados. Se ve aquí la importancia que tiene el disponer de un fondo homogéneo y una iluminación correcta que evite los reflejos y sombras en los contornos de los objetos como se ha visto en el Capítulo 2. De todas formas, se tratarán brevemente algunas técnicas para el tratamiento de texturas, para aquellos casos en que los niveles de intensidad de los objetos no son muy uniformes.

De todo lo expuesto en los párrafos anteriores, se puede llegar a la conclusión de que la etapa de segmentación es, en sí, la verdadera etapa de reconocimiento debido a que la consecuencia de ésta etapa es tener los objetos perfectamente ubicados en la escena.

Antes de comenzar a estudiar las técnicas de segmentación más comunes, se va a proceder a realizar una breve explicación de algunas transformaciones no explicadas en los capítulos anteriores y que son de mucha utilidad para entender las técnicas que más adelante se explican.

4.1 TRANSFORMACIONES MORFOLÓGICAS

Las transformaciones morfológicas se encargan de cambiar la forma y estructura de los objetos. Estas herramientas matemáticas permiten obtener componentes que dan una idea de la forma y estructura de los objetos que forman la imagen. Además, permiten modificar estas formas para separar los objetos unos de otros, obtener contornos primarios, descomponer formas complejas en formas más simples, obtener contornos dentro de entornos ruidosos, reconstruir elementos distorsionados, etc..

Generalmente esta serie de transformaciones se realizan en imágenes previamente binarizadas (en blanco y negro), aunque también existen¹⁹ estudios de aplicación de estas técnicas en imágenes con niveles de grises. Las transformaciones que se van a tratar corresponden a la morfología matemática cuyo lenguaje de expresión es la teoría de conjuntos.

4.1.1 DILATACIÓN BINARIA

Dados dos conjuntos A y B de Z^2 , la dilatación, denotada por $A \oplus B$, se define como:

$$A \oplus B = \{c \in E^N \mid c = a + b \text{ para todo } a \in A \text{ y } b \in B\} \quad (4.1)$$

Es decir, dada una máscara B (formada por unos y ceros), la dilatación de A por B es el conjunto de todos los desplazamientos de x tales que B y A se solapen en al menos un elemento distinto de cero.

Por ejemplo, dada una máscara B formada por una matriz:

0	1	0
1	1	1
0	1	0

Figura 4.1. Máscara B.

y una figura A formada por una matriz:

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

Figura 4.2. Imagen A.

El proceso de dilatación $A \oplus B$ consiste, para cada punto $A(i,j)$, en solapar la máscara B en A (siendo el punto central de B el (i,j)), ver si algún uno de B coincide con algún uno de A y si es así, poner el punto $A(i,j)$ a uno. Es decir, la dilatación consiste en ir desplazando el elemento estructurante (B) por toda la imagen, y si alguno de los elementos de la matriz (denominados vecinos de la máscara) coincide con un *pixel* del entorno, entonces el *pixel* donde está centrado en A se pone a uno.

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

Figura 4.3. Imagen resultante $A \oplus B$.

Existen otros métodos para realizar la dilatación de los objetos, por ejemplo contando los *pixels* que rodean a uno central y si supera un umbral se pone a uno; pero el más utilizado es el expuesto en este documento.

La dilatación, también llamada “crecimiento”, “llenado”, “expansión”, etc., como se ve en la Figura 4.3, produce un efecto de engrosamiento de los bordes del objeto. Este efecto es muy valorado para aumentar el contorno de los objetos y unir líneas discontinuas de estos, producidas por algún filtrado, etc.. La máscara, dependiendo de la forma de ésta, se puede utilizar para realizar la dilatación en la dirección o direcciones que más interesen.

Como se puede apreciar en la Figura 4.4, la dilatación tiende a unir más los objetos que están pegados uno con otros y a reducir los agujeros interiores de los objetos, aumentando por otro lado el grosor de los mismos.

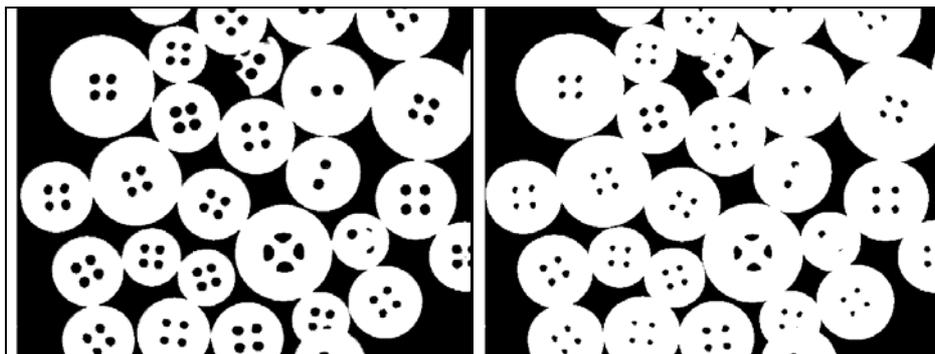


Figura 4.4. Imagen binarizada (izquierda) y dilatación de la misma(derecha)

4.1.2 EROSIÓN BINARIA

Dados dos conjuntos A y B de Z^2 la erosión, denotada por $A \ominus B$, se define como:

$$A \ominus B = \{x \in E^N \mid x = x + b \in A \text{ para todo } b \in B\} \quad (4.2)$$

La erosión es la función dual de la dilatación (pero no es la inversa, es decir, si se hace una erosión y luego una dilatación no se queda la imagen como estaba). Dada una imagen A se erosiona por B cuando para todos los puntos x tales que B , trasladado por x , está contenido en A . Es decir, la erosión pone a cero todos los *pixels* de la imagen que no contengan completamente al elemento estructurante en su entorno.

Si la dilatación expandía los bordes y contornos de los objetos, la erosión reduce los contornos de los objetos. Se utiliza para separar objetos que están unidos por una pequeña parte de sus contornos.

Por ejemplo, usando la máscara del caso anterior:

0	1	0
1	1	1
0	1	0

Figura 4.5. Máscara B.

y una figura A formada por una matriz:

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

Figura 4.6. Imagen A.

El proceso de erosión $A \ominus B$ consiste, para cada punto $A(i,j)$, en solapar la máscara B en A (siendo el punto central de B el (i,j)), y ver si todos los puntos a uno de la máscara B coinciden con los de la imagen, poniéndolo a uno. En caso contrario se pone a cero. Es decir, la dilatación consiste en ir desplazando el elemento estructurante (B) por toda la imagen, y si alguno de los elementos de la matriz (denominados vecinos de la máscara) no coincide con un *pixel* del entorno, entonces el *pixel* donde está centrado en A se pone a cero, si todos coinciden con los de la máscara entonces se pone a uno.

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

Figura 4.7. Imagen resultante $A \ominus B$.

Igualmente que en la dilatación, existen otros métodos para realizar la erosión de los objetos (mediante umbralización), pero el más común es el explicado aquí. La máscara, dependiendo de la forma de ésta, se puede utilizar para realizar la erosión en la dirección o direcciones que nos interesen.

En la Figura 4.8 se puede ver un ejemplo de erosión de una imagen donde se aprecia que los objetos se van separando unos de otros, reduciéndose por otro lado su grosor y diámetro.

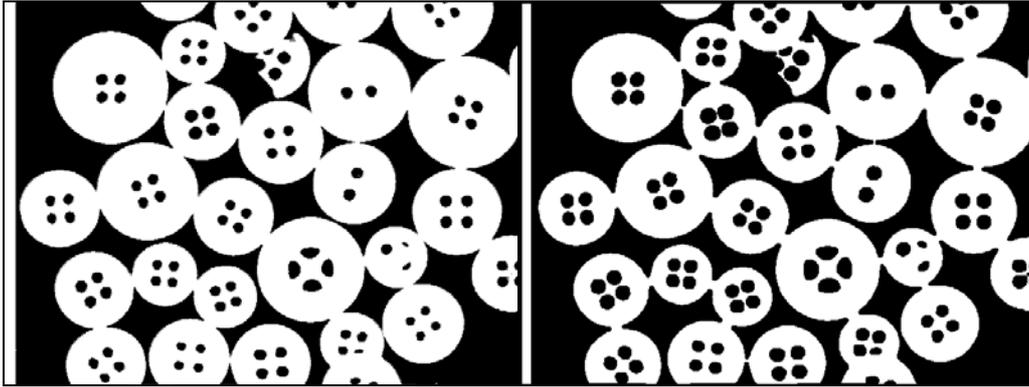


Figura 4.8. Imagen binarizada (izquierda) y erosionada (derecha)

4.1.3 APERTURA

La apertura consiste en realizar en una imagen una erosión y después una dilatación. Aunque a simple vista puede parecer que la imagen va a quedar como está, esto no es así debido a que la dilatación no es una operación inversa de la erosión.

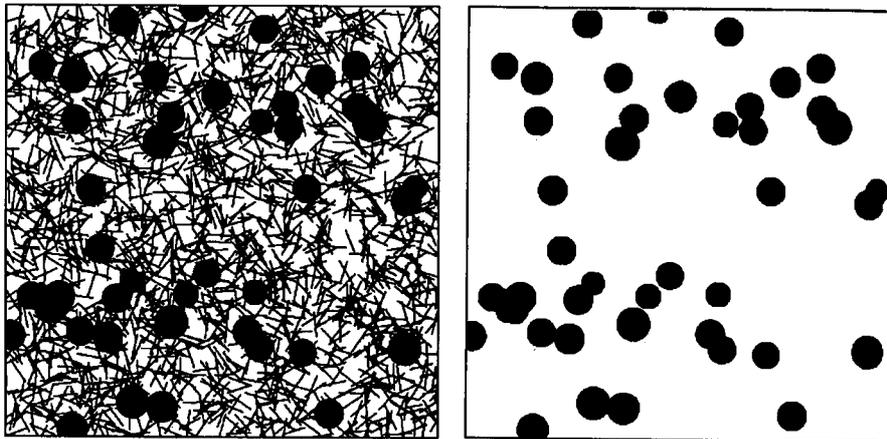


Figura 4.9. Imagen con ruido (izquierda) que se ha filtrado mediante la realización de una apertura aplicando una máscara en forma de disco (derecha).

La apertura es muy utilizada para:

Segmentación de objetos, separando unas formas de otras.

Descomposición de objetos en elementos más simples.

Extracción de formas determinadas en un entorno con ruido (ver Figura 4.9).

Eliminar salientes estrechos.

Separar objetos que no están demasiado pegados.

Aumentar los agujeros que están dentro de los objetos.

Las aperturas pueden estar formadas por un número de erosiones y dilataciones diferentes, produciendo resultados muy dispares.

Existen por otro lado, en la bibliografía especializada²⁰, variaciones especiales de los algoritmos de apertura para por ejemplo: separar objetos sin agrandar los agujeros, etc..

4.1.4 CIERRE

El cierre hace exactamente lo contrario que la apertura, primero se realiza la dilatación y después la erosión.

La combinación de las operaciones de apertura y cierre pueden servir para múltiples operaciones: filtrado, segmentación, etc.; simplemente con jugar con el tipo de máscaras y tamaños.

En la Figura 4.10 se puede ver como simplemente con aperturas y cierres se puede descomponer una figura en su base, tronco y partes horizontales:

De la figura con ruido impulsivo (Figura 4.10.a), se realiza una apertura con una máscara en forma de disco de radio 1 (Figura 4.10.b)

Cerrado con disco de radio 4, se rellenan los huecos dejados en el paso anterior (Figura 4.10.c).

Se eliminan los bordes verticales mediante una apertura de radio 3 (Figura 4.10.d)

Se extrae la base con una máscara formada por un rectángulo (Figura 4.10.e).

La resta de la figura Figura 4.10.d con la Figura 4.10.e determina el residuo (Figura 4.10.f).

Con una máscara formada por una estructura vertical, se obtiene el tronco (Figura 4.10.g).

El uso de las aperturas y cierres suponen un gasto enorme en tiempo de ejecución por parte de un ordenador, por otro lado, la simpleza de sus mecanismos permite la realización directamente en hardware simplificando las tareas de segmentación y reconocimiento para objetos simples en tiempo real.

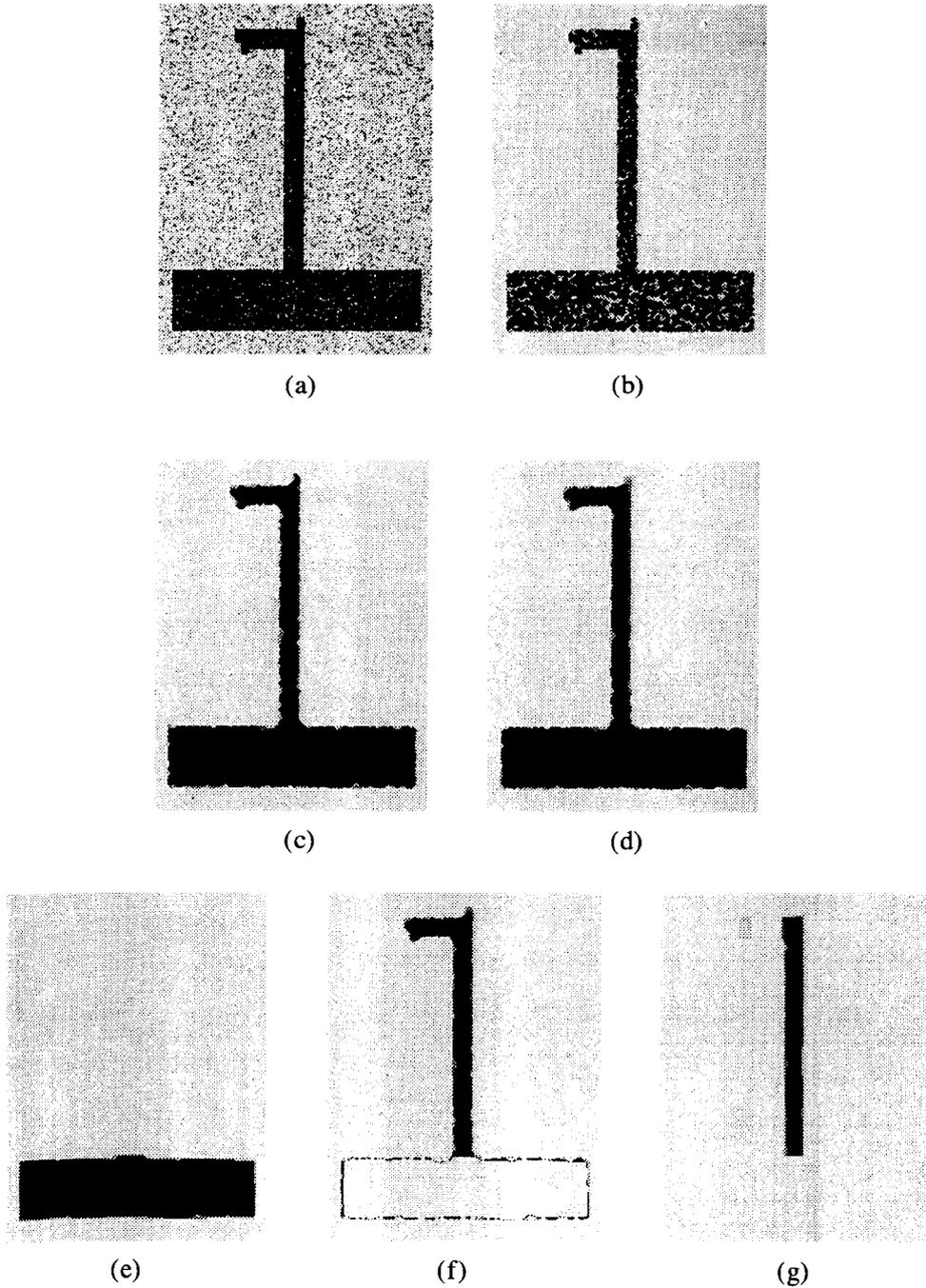


Figura 4.10. Extracción mediante aperturas y cierres de las componentes de una figura: a) imagen original, b) apertura con disco de radio 1, c) cierre con disco de radio 4, d) apertura con disco de radio 3, e) apertura con rectángulo de 21×20 , f) residuo de la apertura anterior, g) apertura con máscara de estructura vertical.

4.1.5 ESQUELETOS

Este tipo de operación reduce los objetos de una imagen, a un esqueleto de grosor un *pixel*. Es decir, dado un objeto cualquiera va reduciendo su grosor hasta dejarlo con un grosor de un pixel.

El esqueleto conseguido da una idea de la forma inicial del objeto. En la Figura 4.11 se puede ver la figura con un grupo de botones y el esqueleto resultante.

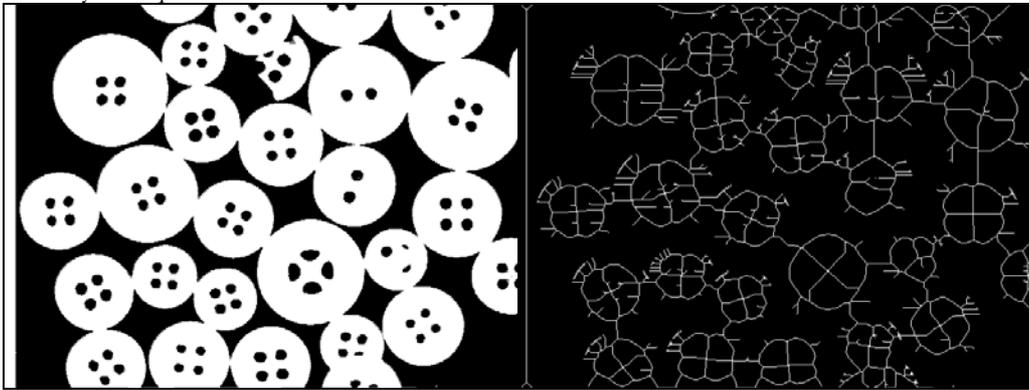


Figura 4.11. Resultado de la esqueletización de la figura izquierda.

El problema de este tipo de transformación, es el tiempo prohibitivo que supone su cómputo. Sólo pudiéndose implementar con hardware específico que acelere el proceso de cálculo.

4.1.6 OTRAS TRANSFORMACIONES

Además de las transformaciones básicas que se han visto en los puntos anteriores, existen otras que permiten manipular imágenes binarias de forma muy sencilla. En la instrucción *bwmorph* de Matlab v5.1 existen múltiples formas de actuar con los puntos de la imagen binaria. Algunos de los más importantes son los siguientes:

'bothat' (bottom hat): Que consiste en realizar un cierre de la imagen y después restar el resultado a la imagen original.

'bridge': Une pixels sin conectar.

'clean': Elimina pixels sueltos.

'fill': Rellena pixels que están a 0 y rodeados de 1.

'diag': Hace un relleno diagonal para eliminar la conectividad 8.

'hbreak': Elimina pixels que están conectados en forma de H.

'majority': Pone un pixel a 1, si 5 o más vecinos (una máscara de 3x3) están a 1.

'dilate': Realiza la dilatación (3x3).

'erode': Realiza la erosión (3x3).

'open': Realiza una apertura (3x3).

'close': Realiza un cierre (3x3).

'remove': Elimina pixels interiores. Pone un *pixel* a cero si los cuatro vecinos están a 1 (con conectividad 4).

'shrink': Reduce objetos a puntos. Si estos tienen agujeros, los convierte a puntos formando un anillo que los une al contorno exterior.

'skel': Realiza la esqueletización.

'spur': Elimina pixels que aparezcan sueltos y solo unidos por una esquina de los mismos.

'thicken': Engorda objetos añadiendo pixels al exterior de los mismos.

'thin': Adelgaza los objetos, quitando pixels de sus contorno.

'tophat': Realiza la apertura de la imagen y se la resta a la imagen original.

Por otro lado, aplicando la matemática de conjuntos se pueden obtener gran cantidad de transformaciones²¹ que pueden ser útiles para diversas acciones: por ejemplo la transformación "hit & miss" (traducido como golpe y pierde) se basa en la combinación de las operaciones de erosión y dilatación, para seleccionar pixels que tiene una cierta geometría, como por ejemplo: esquinas, puntos aislados, puntos de los bordes, etc.

²¹ En *Computer and Robot Vision. Vol I* de ADDISON WESLEY se trata con profundidad toda la matemática morfológica tanto binaria como en escala de grises, y sus aplicaciones en la visión computacional.

4.2 TÉCNICAS DE SEGMENTACIÓN

En el campo de la segmentación, en la actualidad, existe un gran número de investigadores que están desarrollando nuevas técnicas para poder abordar problemas cada vez más complejos y que hace unos años no se podían resolver.

En el presente trabajo, se han estudiado las técnicas más comunes que tienen cabida en la aplicación y desarrollo de sistemas de control de calidad con entornos más o menos controlados.

Si en el desarrollo de una tarea de visión, se advierte que no se puede resolver la etapa de segmentación con las tareas descritas, será conveniente replantearse todas las etapas anteriores que la simplifiquen (adquisición, preprocesado, etc.) o buscar técnicas de segmentación más sofisticadas.

4.2.1 SEGMENTACIÓN BASADA EN UMBRALIZACIÓN

El histograma, como se ha visto en capítulos anteriores, permite ver la frecuencia relativa de aparición de cada nivel de luminancia de una imagen.

El proceso de umbralización basado en el histograma, consiste en agrupar los pixels según sus niveles de intensidad luminosa. Por ejemplo, si se analiza la imagen de una manzana roja en un fondo blanco, el histograma estará formado por dos montañas correspondientes a los pixels del nivel de gris (si la imagen obtenida es en niveles de grises) y al fondo blanco. En la Figura 4.12 se puede ver la imagen de un botón y su histograma correspondiente.

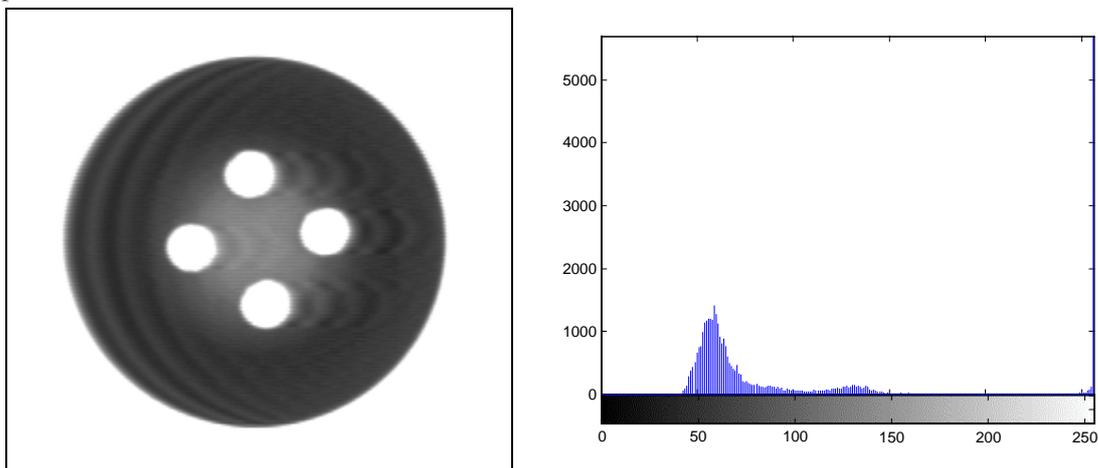


Figura 4.12. Imagen de un botón y su histograma correspondiente.

En el histograma de la Figura 4.12 se puede distinguir fácilmente la montaña correspondiente a los niveles de gris del botón y a la izquierda del todo una línea que indica los niveles de luminosidad del fondo color blanco nieve (cercanos al 255).

Si en vez de tener un objeto, se tiene varios con diferentes niveles de grises, aparecerían varios montículos en el histograma correspondientes a cada objeto.

La segmentación, en este caso, consiste en encontrar el umbral necesario para realizar una binarización que separe el objeto (poniéndolo en blanco) del fondo (en negro o viceversa).

El problema es encontrar el valor del umbral que separe fácilmente los objetos del fondo. Esto puede resultar sencillo o complicarse en demasía. Si las condiciones de iluminación y del fondo son homogéneas y constantes, el umbral se puede realizar de forma manual mediante simple experimentación.

Si la iluminación, fondo o los objetos pueden cambiar, será necesario desarrollar un sistema de cálculo del umbral de forma automática. Existen diferentes formas de obtener este umbral.

MÉTODO P-CUANTIL

Este método usa el conocimiento sobre el área o el tamaño del objeto que se desea extraer para segmentar la imagen. Si se supone que en una aplicación determinada los objetos ocupan alrededor del p por ciento del

tamaño del área de la imagen. Utilizando este conocimiento es posible particionar la imagen de modo que asigne el p por ciento de los pixels al objeto.

Es claro que el uso de este método es muy limitado. Sin embargo, en aplicaciones como preprocesamiento para reconocimiento de caracteres es muy útil.

UMBRALIZACIÓN BASADA EN LA BÚSQUEDA DE MÍNIMOS

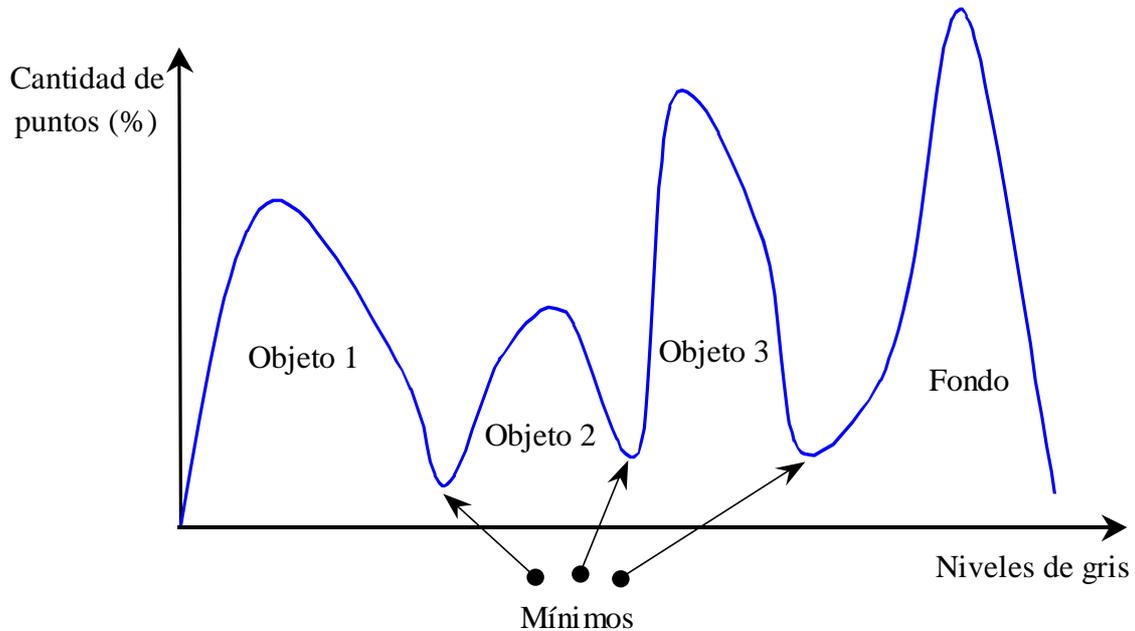


Figura 4.13. Histograma de una figura con tres objetos y un fondo blanco

Dado que un objeto con unos niveles de grises homogéneos va a tener en su histograma un montículo determinado, el procedimiento consiste en obtener los mínimos de esos valles y asignarlos como valores iniciales de los umbrales buscados.

Como se ve en la Figura 4.13, para poder obtener uno de los objetos simplemente habrá que umbralizar entre los dos mínimos correspondientes al montículo que forma. Por ejemplo, si se quiere obtener el objeto 2 se tendrán que obtener los dos mínimos (anterior y posterior) y después binarizar con dos umbrales.

La forma de realizar la binarización con dos umbrales, consistirá en poner a cero todos los *pixels* cuyo nivel de luminancia sea menor que el primer umbral y mayor que el segundo, y poner a uno los puntos cuyo nivel de luminancia esté entre el primer y segundo umbral (los puntos correspondientes al objeto a segmentar).

En la práctica, la obtención de los mínimos no es tan sencilla:

Primero por que el histograma presenta un montón de mínimos debidos al ruido que será necesario filtrar con un filtro paso bajo para alisarlo convenientemente de forma que sólo queden los mínimos realmente correspondientes a los valles de los objetos.

Segundo, porque generalmente los histogramas de varios objetos se pueden mezclar, dado que éstos pueden tener puntos con niveles de luminancia muy próximos. Si esto ocurre el método de umbralización basado en histograma no será muy adecuado, siendo necesaria la búsqueda de otras técnicas que solucionen el problema.

UMBRALIZACIÓN BASADA EN LA MINIMIZACIÓN DE LA VARIANZA DE CADA GRUPO DE VALORES

Para cada umbral t asignado a un histograma se puede determinar que éste lo separa en dos grupos: aquellos *pixels* con niveles de grises menores o iguales que t y aquellos mayores de t . Es decir, cada umbral determinará dos varianzas, una para cada grupo de puntos obtenidos.

La definición para el mejor umbral, según Otsu (1979), es aquél cuya suma de pesos de cada grupo de varianzas se minimiza. Siendo los pesos las probabilidades respectivas de cada grupo.

Al final se trata de minimizar²² la función:

$$\sigma_w^2(t) = q_1(t) \cdot \sigma_1^2(t) + q_2(t) \cdot \sigma_2^2(t) \quad (4.3)$$

Siendo σ_w^2 la suma de las varianzas de los dos grupos (σ_1^2 y σ_2^2), y $q_1(t)$ y $q_2(t)$ las probabilidades para los grupos 1 (menores o iguales a t) y el grupo 2 (mayores a t).

El desarrollo matemático que permite el cálculo del umbral que minimiza esta función, viene explicado con profundidad en la bibliografía indicada al final del documento.

UMBRALIZACIÓN BASADA EN TÉCNICAS DE RECONOCIMIENTO DE FORMAS

Esta técnica consiste en aproximar los lóbulos, que aparecen en un histograma debido a los objetos, en funciones de campana de Gauss, típicas de las distribuciones de luminancias más habituales.

Sugerido por Kittler e Illingworth (1985) tiene un comportamiento mejor que el anterior, pudiéndose aplicar al histograma de varios objetos ajustando a cada uno de ellos una función de Gauss.

Las técnicas de umbralización del histograma aplicadas a toda la imagen a la vez, no suelen ser muy prácticas ya que el cambio de la iluminación a lo largo de la imagen hace que los resultados no sean satisfactorios.

En vez de realizarse un histograma de una imagen completa, es más práctico dividir la imagen en subzonas disjuntas y calcular el histograma de cada una de ellas, actuando localmente según los resultados que dé el histograma.

También se puede aplicar una ventana para cada *pixel* y calcular el histograma para modificarlo. Realmente la carga computacional puede ser necesaria para realizar este último proceso puede ser prohibitiva.

UMBRALIZACIÓN EN ESTRUCTURAS DE DATOS JERÁRQUICAS

La idea de la umbralización utilizando estructuras de datos jerárquicas está basada en los métodos de umbralización locales. La idea es detectar la presencia de una región en una imagen de baja resolución y dar a la región más precisión en imágenes de altas a total resolución. Algunos métodos trabajan sobre estructuras de datos piramidales, donde las imágenes de baja resolución se calculan a partir de las de alta resolución utilizando promedios de niveles de gris. Los métodos más sencillos comienzan en la imagen de menor resolución (la de nivel mayor en la pirámide) aplicando cualquiera de los métodos de segmentación hasta ahora discutidos. El paso siguiente produce mejor precisión en la segmentación: los pixels próximos a las fronteras son resegmentados en fondo u objeto. Este crecimiento en la precisión se repite para cada par de niveles de la pirámide hasta llegar al nivel de resolución total en el que se obtiene la segmentación total. Una gran ventaja de este método es la enorme reducción del nivel de ruido puesto que las segmentaciones iniciales están basadas en versiones alisadas de la imagen.

Otra alternativa es buscar pixels *significativos* en la imagen y segmentar la imagen en regiones de tamaño apropiado. La estructura piramidal se usa de nuevo bien con tamaño 2x2 o 4x4, en este segundo caso con solapamiento. Un detector de pixels importantes se aplica a todos los pixels de la pirámide. Este detector se basa en ventanas de tamaño 3x3 y responde si el pixel central difiere de la mayoría de los pixels en su entorno. Se supone entonces que la existencia de pixels significativos viene dada por la presencia de regiones diferentes en la imagen con total resolución. La parte de la imagen original que ha producido este entorno 3x3 es entonces umbralizada. El umbral se determina como la media del nivel de gris del pixel significativo (media de la región) y la media de sus ocho vecinos (media del fondo).

Aunque en esta sección se han discutido las principales aproximaciones a la detección de fronteras basada en la umbralización, existen muchas variaciones que no se han mencionado. El lector interesado puede consultar la literatura especializada²³.

22 Haralick Robert M., Shapiro Linda G.. *Computer and Robot Vision. Vol I.* ADDISON WESLEY.

23 M. Sonka, V. Hlavac, R. Boyle. *Image Processing, Analysis and Machine Vision.* Chapman & Hall Computing.
R. C. Gonzalez, R.E. Woods. *Digital Image Processing.* Addison Wesley.
R. Kasturi R. Jain, B.G. Schunck. *Machine Vision.*

El artículo "Survey of thresholding techniques" de A.K.C. Wong P.K. Sahoo, S. Soltani, Y.C. Chen en *Computer Vision, Graphics and Image Processing* es una buena recopilación de métodos de segmentación basados en umbralización.

4.2.2 TÉCNICAS DE SEGMENTACIÓN BASADAS EN BORDES

Históricamente el primer grupo de métodos de segmentación, que es todavía hoy muy importante es el basado en información sobre bordes en la imagen. Las técnicas de segmentación basadas en bordes se basan en los bordes encontrados en la imagen por los detectores. Los métodos de extracción de bordes analizados en capítulos anteriores no pueden ser utilizados como un resultado de segmentación. Otros pasos posteriores han de seguir para combinar los bordes en cadenas de bordes que corresponden con fronteras de objetos en la imagen. El objetivo final es alcanzar al menos una segmentación parcial, es decir, agrupar bordes locales en una imagen donde sólo cadenas de bordes con una correspondencia con objetos en la imagen o partes de la imagen están presentes.

Existen diferentes métodos para la segmentación basada en bordes que difieren en la estrategia para la construcción de la frontera, así como en la cantidad de información previa que se incorpora en el método. Es obvio, que a mayor información mejor segmentación. En este capítulo se discutirán sólo métodos que incorporan muy poca información en el proceso de segmentación²⁴.

RELAJACIÓN DE BORDES

Los bordes que se obtienen de los métodos descritos en las secciones anteriores están usualmente muy afectados tanto por el ruido de la imagen como por la no presencia de partes que deberían estar presentes. Si en el proceso de paso de borde a frontera se tienen en cuenta las propiedades de los bordes vecinos presentes, es posible aumentar la calidad de los resultados. Todas las propiedades de una imagen junto con la posibilidad de existencia de más bordes, se evalúan iterativamente con más precisión hasta que el contexto de los bordes queda claro. Basado en la fuerza de bordes en una vecindad local especificada, la confianza de cada borde se aumenta o disminuye²⁵. Un borde débil posicionado entre dos bordes fuertes proporciona un ejemplo de contexto; es altamente probable que este borde débil entre ellos sea parte de una frontera. Si, por otra parte, un borde, incluso uno fuerte no tiene contexto que lo soporte, puede que no sea parte de ninguna frontera.

El método discutido aquí es un ejemplo clásico de evaluación de contexto en bordes. El método utiliza bordes tipo grieta (entre pixels). El contexto del borde se considera a ambos lados de un borde, e , con el sistema de vecinos que se muestra en la Figura 4.14.

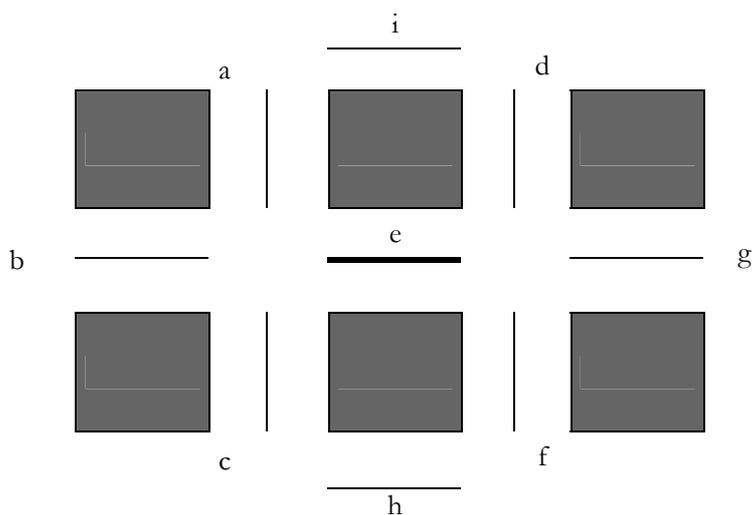


Figura 4.14. Borde tipo grieta e con sus vecinos.

²⁴ Métodos más complejos y también más efectivos son discutidos en: M. Sonka, V. Hlavac, R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman & Hall Computing.

²⁵ R.A. Hummel A. Rosenfeld, S.W. Zucker. "Scene labelling by relaxation techniques", *I.E.E.E. Trans. on Systems, Man and Cybernetics*.
S.W. Zucker. "Relaxation labelling, local ambiguity, and low level vision", *Pattern Recognition and Artificial Intelligence*. Academic Press.
E.M. Riseman, M.A. Arbib. "Computational techniques in the visual segmentation of static scenes", *Computer Graphics and Image Processing*.
E.R. Hancock, J. Kittler. "Edge labelling using dictionary-based relaxation", *I.E.E.E. Trans on Pattern Analysis and Machine Intelligence*.

Todas las posibles posiciones de los vecinos de e deben ser incluidas para cubrir la forma en la que se continua en el entorno. Además dos posiciones paralelas con e pueden ser incluidas en el sistema de vecinos, ya que estas posiciones paralelas compiten con e en la colocación del borde. La relajación de bordes busca la construcción de fronteras continuas de forma que se discutirán los patrones que se pueden encontrar en el entorno. Dado un borde central e , éste tiene un vértice a cada uno de sus lados y cada uno de estos vértices tiene tres posibles formas de continuar. La forma de continuar de un vértice se evalúa de acuerdo al número de bordes que de él emanan, sin contar e , y este número define el tipo de vértice. Entonces el tipo de borde e puede ser representado mediante $i - j$, donde i notará la configuración de bordes que se produce a su izquierda (en su vértice izquierdo) y j el que hay a la derecha. Por ejemplo, se asignará el tipo 0 - 0 a la configuración que se muestra en la Figura 4.15.a, el 2 - 0 para la Figura 4.15.c, etc.. Es obvio que por simetría sólo se deberá considerar el caso $i \leq j$. Se tienen las siguientes situaciones de contexto:

- 0 - 0 borde aislado, influencia negativa en la confianza del borde.
- 0 - 2, 0 - 3 final sin salida, influencia negativa en la confianza del borde.
- 0 - 1, incierto, poco positivo o no influencia.
- 1 - 1, continuación, influencia fuertemente positiva en la confianza del borde.
- 1 - 2, 1 - 3 continuación a la intersección de bordes, influencia medio positiva sobre la confianza del borde.
- 2 - 2, 2 - 3, 3 - 3, puente entre fronteras, no es necesario para la segmentación no tiene influencia en la confianza del borde.

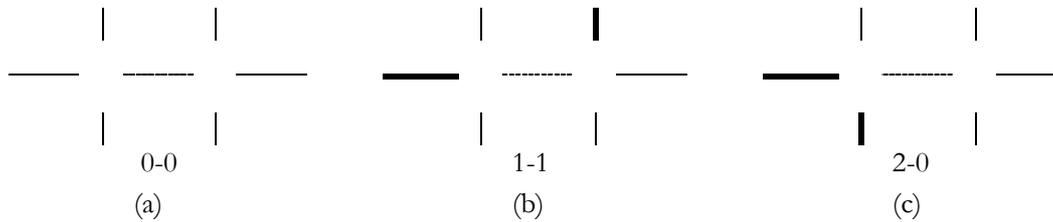


Figura 4.15. Borde tipo grieta e con sus vecinos.

Una relajación de bordes es un método iterativo, en la que la confianza de los bordes tiende a cero (desaparición del borde) o a uno (los bordes forman una frontera). La confianza $c^{(1)}(e)$ de cada borde e en la primera iteración puede definirse como una magnitud normalizada del borde tipo grieta, con una normalización basada bien en el máximo global de los bordes tipo grieta de toda la imagen, o en el máximo local de algún entorno grande del borde, en este último caso el objetivo es disminuir la influencia de unos pocos valores muy altos de la magnitud del borde en la imagen.

El algoritmo funcionaría como sigue:

Evaluar la confianza $c^{(1)}(e)$ para todos los bordes tipo grieta de la imagen.

Encontrar el tipo de borde de cada borde basado en la confianza de los bordes de sus vecinos.

Actualizar la confianza $c^{k+1}(e)$ de cada borde e de acuerdo con su tipo y su confianza previa.

Parar si todas las confianzas de los bordes han tendido a cero o uno. Repetir los pasos (2)-(3) en caso contrario.

Los pasos principales del algoritmo anterior son la evaluación de los tipos de vértices seguido de la evaluación del tipo de borde y la forma en que se modifica la confianza en el borde. Un vértice es considerado de tipo i si:

$$\text{tipo}(i) = \max_k(\text{tipo}(k)), \quad k = 0, 1, 2, 3$$

$$\text{tipo}(0) = (m - a)(m - b)(m - c)$$

$$\text{tipo}(1) = a(m - b)(m - c)$$

$$\text{tipo}(2) = ab(m - c)$$

$$\text{tipo}(3) = abc$$

donde a, b, c son los valores normalizados de los otros bordes y, sin pérdida de generalidad, se puede suponer que $a \geq b \geq c$; q es una constante para la que un valor de 0.1 parece apropiado y $m = \max(a, b, c, q)$. Obsérvese que la introducción de la cantidad q asegura que $\text{tipo}(0)$ es no cero para pequeños valores de a .

Por ejemplo, si se elige $q = 0.1$, un vértice $(a, b, c) = (0.5, 0.05, 0.05)$ es un vértice de tipo uno, mientras que un vértice $(0.3, 0.2, 0.2)$ es un vértice tipo tres. Los mismos resultados se pueden obtener observando el número de bordes de un vértice que están por encima de un umbral. El tipo de borde se encuentra como la concatenación del tipo de vértices y la confianza de bordes se modifica como sigue:

$$\text{Aumento de confianza: } c^{k+1}(e) = \min(1, c^k(e) + \delta)$$

Disminución de confianza: $c^{k+1}(e) = \max(0, c^k(e) - \delta)$

donde δ es una constante apropiada, usualmente en el rango [0.1, 0.3].

Las reglas para la modificación de la confianza del borde pueden simplificarse y usar un único δ , sin incluir el crecimiento / decrecimiento débil, moderado o fuerte. Además, los vértices tipo 2 y 3 pueden considerarse del mismo tipo de implementación.

La relajación de bordes, tal y como ha sido descrita, mejora rápidamente el etiquetado original de bordes en unas pocas iteraciones. Sin embargo con muchas iteraciones empeora el resultado. La razón para esta conducta extraña está en la búsqueda de un máximo global del criterio de consistencia sobre toda la imagen, lo cual puede no ser óptimo localmente. Una solución es asignar a la confianza de un borde cero si está por debajo de un umbral y uno si está por encima de otro. Por tanto, se añadiría un paso adicional a la computación de la confianza:

si $*1cm^{k+1}(e) > T_1 *1c$ asignar entonces $*1cm^{k+1}(e) = 1$

si $*1cm^{k+1}(e) < T_2 *1c$ asignar entonces $*1cm^{k+1}(e) = 0$

donde T_1 y T_2 son parámetros que controlan la convergencia del algoritmo.

EXTRACCIÓN DE LA FRONTERA

Las fronteras son bordes unidos que caracterizan la forma de un objeto. Son, por tanto, útiles para calcular rasgos geométricos como tamaño u orientación.

Conectividad

Conceptualmente, las fronteras pueden encontrarse trazando los bordes conectados (bien sea 4 u 8 conectados). Sin embargo, hay dificultades asociadas con estas definiciones de conectividad, como muestra la Figura 4.16. Si se utiliza la 4-conectividad, los segmentos 1, 2, 3 y 4 serían clasificados como disjuntos, aunque obviamente parecen formar un anillo. Bajo la 8-conectividad estos segmentos estarían conectados, pero también lo estaría el interior con el exterior del anillo. En principio, esto podría resolverse utilizando tipos de conectividad distinta para los objetos y para el fondo, pero, obviamente hay otros problemas.

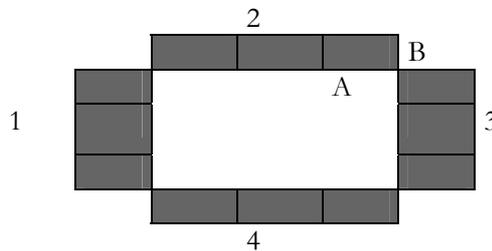


Figura 4.16. Problemas con la conectividad.

Seguimiento de contorno

Como el nombre sugiere, los algoritmos de seguimiento de contorno trazan las fronteras ordenando los puntos de tipo borde sucesivos. Un algoritmo muy utilizado para determinar los puntos del contorno de un objeto es el **algoritmo de la tortuga**. El algoritmo realiza los siguientes pasos:

Se empieza desde un punto cercano al objeto.

Se avanza pixel a pixel hasta encontrar un pixel negro correspondiente al contorno a determinar. En la Figura 4.17 se puede apreciar que el objeto está formado por puntos negros.

Empieza el algoritmo de la tortuga que consiste en avanzar de la siguiente manera: si está en un punto negro gira 90° en SENTIDO CONTRARIO A LAS AGUJAS DEL RELOJ y avanza un pixel, si en cambio está en un punto blanco gira 90° en SENTIDO DE LAS AGUJAS DE RELOJ y avanza un pixel.

Se van almacenando las coordenadas de todos los pixels negros que se van encontrando.

El algoritmo termina cuando llega de nuevo al primer punto negro encontrado.

Como en el vector del contorno aparecen algunos puntos repetidos (algunas esquinas), se eliminan.

En la Figura 4.17 se puede observar que este algoritmo se desplaza bordeando todo el objeto (siempre que sea un objeto cerrado y no se salga de la imagen).

Una vez obtenido el vector con el contorno del objeto, se puede analizar este para determinar gran cantidad de parámetros.

Por otro lado, se puede ver la importancia de que el objeto esté debidamente binarizado sin “cortes” ni ruidos que puedan interferir la obtención correcta de su contorno.

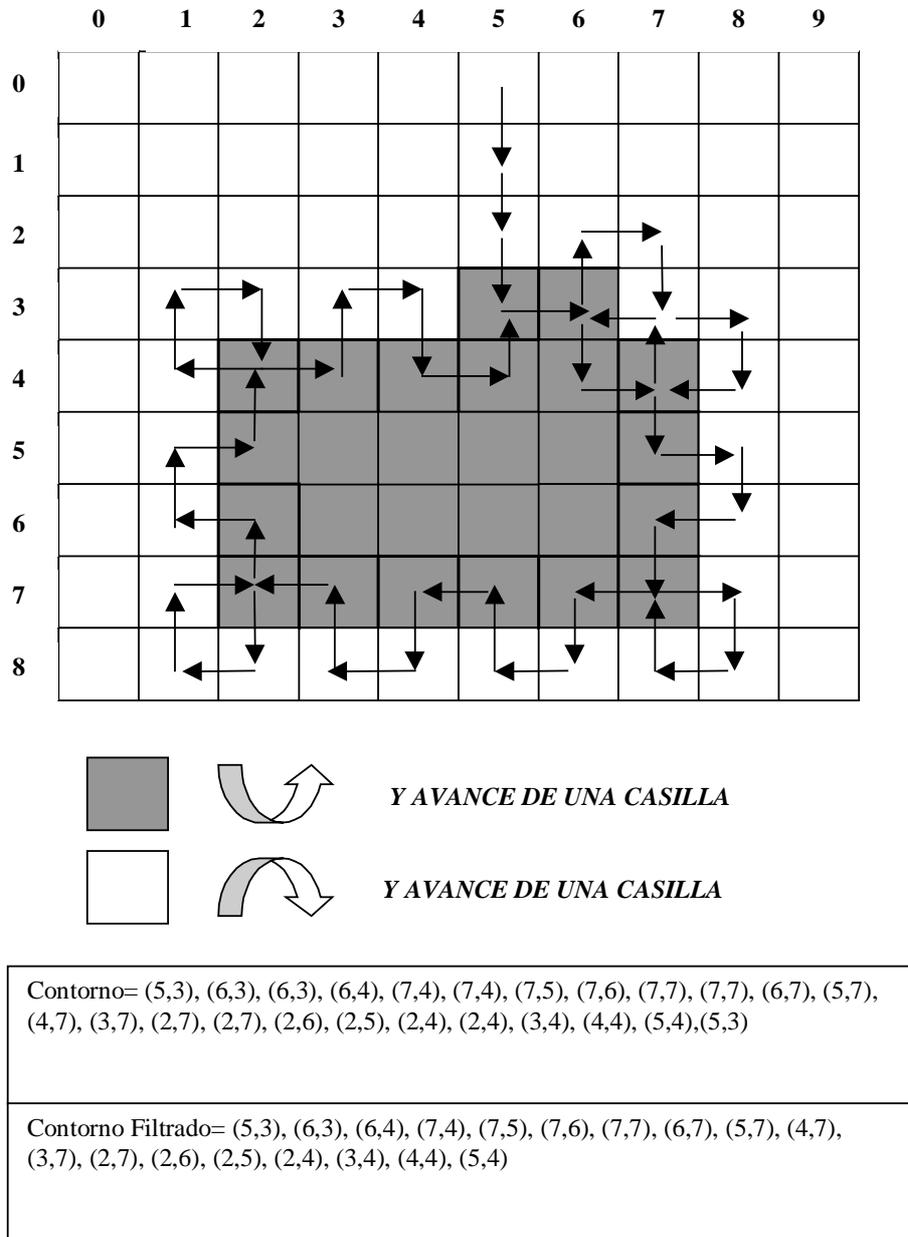


Figura 4.17. Funcionamiento del algoritmo de la tortuga.

Unión de bordes y búsqueda heurística en grafos

Una frontera puede ser también vista como un camino que se forma a través de un grafo construido uniendo los elementos bordes. Las reglas de unión dan la forma para conectar los bordes. Supóngase que un grafo con nodos en las posiciones x_k , $k = 1, 2, \dots$ se forma desde el nodo A a B , y que se tiene una función de evaluación $\phi(x_k)$ que da el valor del camino de A a B cuando pasa por el nodo x_k . En algoritmos de búsqueda heurística, se examinan los sucesores del nodo de salida A y se selecciona el que maximiza $\phi(\cdot)$. El nodo seleccionado se convierte ahora en el nodo de salida y se repite el proceso hasta que se alcanza B . La sucesión de nodos seleccionada constituye la frontera.

Obsérvese que el algoritmo no produce un óptimo global. Por ejemplo, se considera una matriz de tamaño 3×5 cuya magnitud de gradiente y dirección de contorno tangencial se muestran en la Figura 4.18. Un pixel X se considera unido a Y si las correspondientes direcciones de contorno forman un ángulo menor de 90° (ver Figura 4.18). Supóngase que la función de mérito utilizada $\phi(x)$ es la suma de los gradientes hasta llegar a x . Se observa que con este modelo, el camino óptimo sería $ADEFB$, sin embargo el óptimo global es $ACDEFB$.

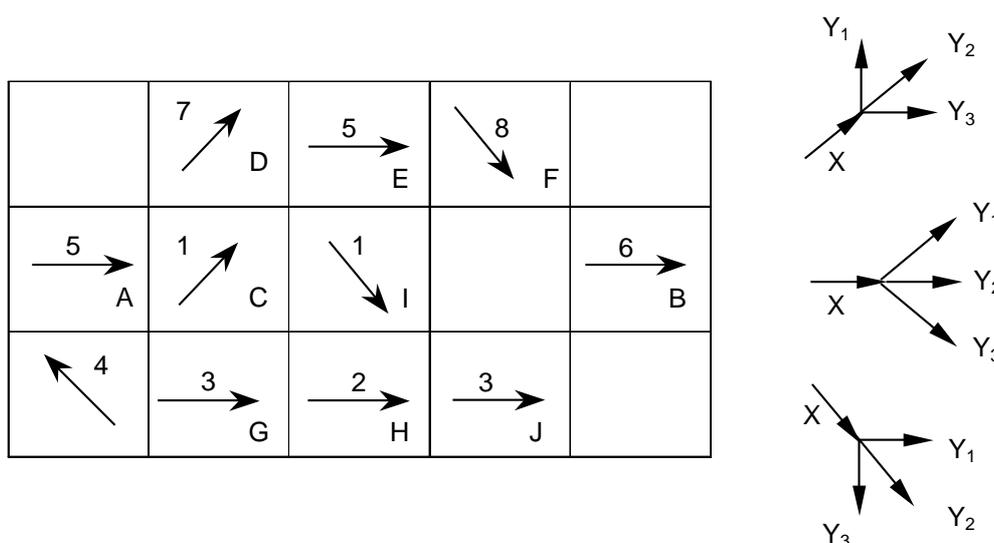


Figura 4.18. Unión heurística de bordes.

TRANSFORMADA DE HOUGH

La transformada Hough, sirve para vectorizar una imagen de líneas y contornos. Es decir, dada una imagen de la que se han extraído sus contornos se procede a convertir estos en una serie de vectores.

Este tipo de transformada no obtiene una serie de líneas que definen los objetos. El problema es que el cálculo de la transformada de Hough es muy lento, debido a que ésta tiene que desplazarse por todos los contornos ajustando las líneas buscadas a éstos.

TRANSFORMADA RADON

Otro tipo de transformada es la transformada Radon²⁶. Este tipo de transformada consiste en proyectar todos los puntos de la imagen en una línea diagonal que va girando de 0 a 180 grados. Es decir, para cada grado se tiene un vector que representa a la recta con esa inclinación y el número de puntos que han incidido en ella. Donde mayores valores se obtengan significará que hay una recta perpendicular a esa inclinación.

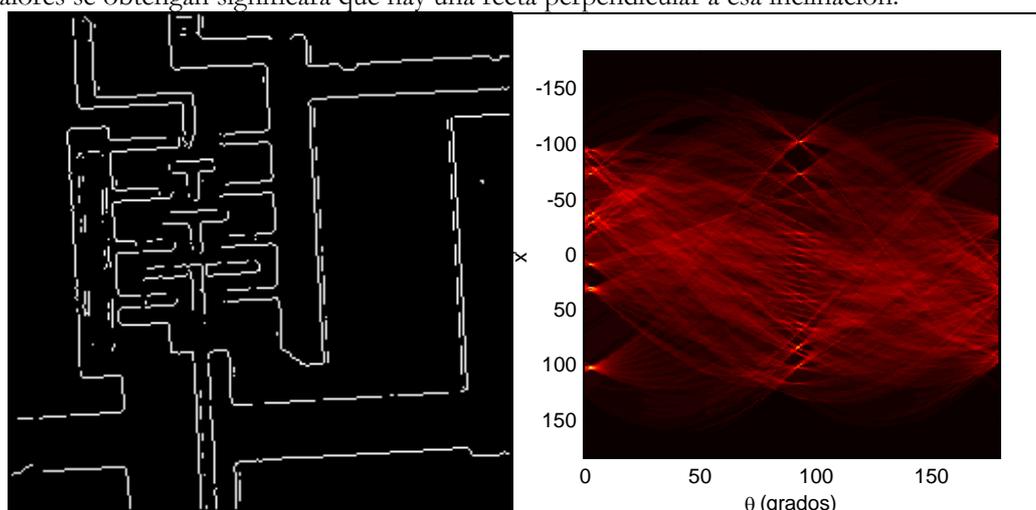


Figura 4.19. Contorno de la imagen de un circuito integrado y la transformada Radon.

Se puede observar los puntos más fuertes que corresponden a las líneas que están a 4° y a 94°.

TRANSFORMADA DEL RELLENADO CRECIENTE

Consiste en asignar un valor a cada pixel según la distancia a la que esté del borde. Esto permite distinguir fácilmente los objetos de una imagen.

En la Figura 4.20 se puede ver como el algoritmo, a partir del borde del objeto va asignando mayores valores a medida que se va separando del centro. Es un método que sirve de continuación a la extracción de los bordes, y que da una idea de la forma del objeto.

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

Figura 4.20. Imagen resultante $A \oplus B$.

4.2.3 SEGMENTACIÓN ORIENTADA A REGIONES

El objeto de los métodos de segmentación descritos en la sección anterior fue encontrar bordes entre regiones. Los métodos que se examinan ahora construyen las regiones directamente.

Las regiones resultantes de estos procesos de segmentación deberán ser homogéneas y maximales, donde maximal significa que los criterios de homogeneidad no se cumplirán tras la unión de una región con alguna de sus adyacentes.

El criterio más simple de homogeneidad utiliza el nivel de gris medio de la región, algunas propiedades simples o un vector de medias para imágenes multibanda.

UNIÓN DE REGIONES

El método más natural para el crecimiento de las regiones es comenzar considerando cada pixel de la imagen como una región, obviamente en este caso cada región es homogénea, pero no necesariamente cumple el criterio de ser las regiones maximales. El proceso habrá de repetirse hasta que las regiones sean maximales. En algoritmo, el método sería:

Definir una segmentación inicial que cumpla el criterio de homogeneidad.

Definir un criterio para unir regiones adyacentes.

Unir las regiones adyacentes si cumplen el criterio de unión. Parar cuando no puedan unirse dos regiones sin romper el criterio de homogeneidad.

En general, podría decirse que lo más importante es seleccionar el criterio para realizar la unión. Algunos criterios de tipo heurístico son:

Unir dos regiones R_i y R_j si $w/P_m > \theta_1$, donde $P_m = \min(P_i, P_j)$, P_i y P_j son los perímetros de R_i y R_j y w es el número de localizaciones de fronteras débiles, (pixels que a ambos lados tienen una diferencia menor que un cierto umbral)

CRECIMIENTO DE REGIONES POR AGREGACIÓN DE PIXELS

Como es obvio el crecimiento de regiones es un procedimiento que agrupa pixels o subregiones en regiones mayores. La forma más sencilla de este proceso es la llamada agregación de pixels que comienza con un conjunto de pixels semilla y a partir de ellos hace crecer la región añadiendo a dichos pixels semilla aquellos vecinos que tienen propiedades similares (nivel de gris, color, textura).

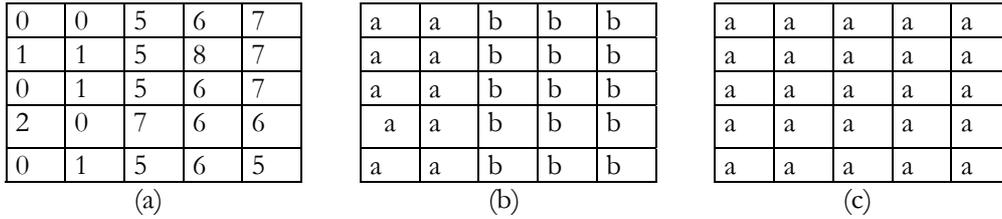


Figura 4.21. Ejemplo de crecimiento de regiones a partir de puntos semilla.

Por ejemplo, véase Figura 4.21, donde los números dentro de las celdillas representan niveles de gris. Sean los puntos de coordenadas (3,2) y (3,4) las semillas consideradas. Utilizando estos dos puntos semilla se obtendrá una segmentación que consiste en dos regiones a lo más: una región R_1 asociada con la semilla (3,2) y otra R_2 asociada con (3,4). La propiedad P usada para incluir un pixel en una región es que la diferencia en valor absoluto entre los niveles de gris del pixel y la semilla sea menor que un umbral T . Cualquier pixel que cumpla la propiedad dada con relación a los dos pixels es asignado arbitrariamente a una de las regiones. La Figura 4.21.b muestra el resultado para $T = 3$. Las dos regiones segmentadas se notan a y b . Con $T = 8$ se obtendría una única segmentación como se muestra en la Figura 4.21.c.

Obviamente el método descrito presenta algunos inconvenientes como la selección de la semilla inicial y la del criterio de similitud. Un problema adicional es la determinación de un criterio de parada. No sólo se debería tener en cuenta en el criterio de parada el nivel de gris, conceptos como forma, tamaño, etcétera son también importantes.

ALGORITMOS DE CRECIMIENTO CON MEZCLADO

El algoritmo anterior sólo asigna a cada *pixel* una zona, pero es muy probable que existan puntos que pertenezcan a varias zonas.

Un mecanismo de crecimiento con mezclado se puede incorporar, utilizando la media aritmética m_i y la desviación estándar de una región de n puntos:

$$m_i = \frac{1}{n} \cdot \sum_{(k,l) \in R_i} f(k,l) \quad (4.4)$$

$$\sigma_i = \sqrt{\frac{1}{n} \cdot \sum_{(k,l) \in R_i} (f(k,l) - m_i)^2} \quad (4.5)$$

De forma que para decidir si dos regiones R_1 y R_2 deben mezclarse se considera la ecuación:

$$|m_1 - m_2| < k \cdot \sigma_i \quad i = 1,2 \text{ (AIII.6)}$$

DIVISIÓN Y FUSIÓN DE REGIONES

Los procedimientos que se acaban de describir hace crecer las regiones a partir de un conjunto de puntos semilla. Una alternativa es subdividir una imagen inicialmente en un conjunto de regiones disjuntas y luego unir las o separarlas en un intento de que:

- todas ellas cubran la imagen completa,
- cada una de ellas esté conectada,
- sus intersecciones dos a dos sea el conjunto vacío,
- algún predicado de homogeneidad sea verdadero en cada región,
- la unión de dos cualesquiera de ellas no cumpla algún predicado de homogeneidad.

Sea R la imagen entera y P un predicado de homogeneidad. Supuesto que se tiene una imagen cuadrada, una aproximación para segmentar R es subdividir la imagen sucesivamente en cuadrantes menores, de forma que se tendrá que para cada región R_i , $P(R_i) = V$. Esto es, si $P(R) = F$, se dividirá el cuadrante en subcuadrantes y se repetirá el proceso con los subcuadrantes obtenidos. Esta técnica de división puede representarse muy bien a través de los llamados quad-trees, esto es, un árbol en el que cada nodo tiene cuatro descendientes (ver Figura

4.22). Obsérvese que la raíz del árbol corresponde a la imagen original y que cada nodo es una subdivisión. En este caso, sólo R_4 se dividió más.

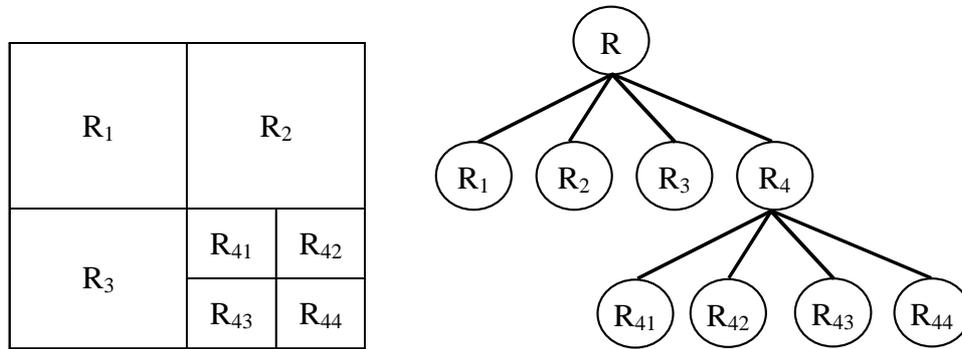


Figura 4.22. Ejemplo de Quad-tree.

Si sólo se realizaran procesos de separación se podría terminar con regiones idénticas que fueran adyacentes pero no identificadas como una misma región. Este inconveniente puede resolverse si además de la separación se permite la unión. La idea es unir regiones adyacentes cuyos pixels combinados satisfagan el predicado P , esto es dos regiones adyacentes R_j y R_k son unidas cuando y sólo cuando $P(R_j \cap R_k) = V$.

La discusión anterior puede resumirse en el procedimiento siguiente:

Dividir en cuatro regiones disjuntas cualquier región R_i para la que $P(R_i) = F$,

Unir cualesquiera dos regiones adyacentes R_j y R_k para las que $P(R_j \cap R_k) = V$,

Parar cuando no sea posible realizar mas uniones o divisiones.

Es una técnica muy utilizada, cuando se pretende reconocer objetos por su forma más tosca, ya que este tipo de técnica elimina los detalles del objeto.

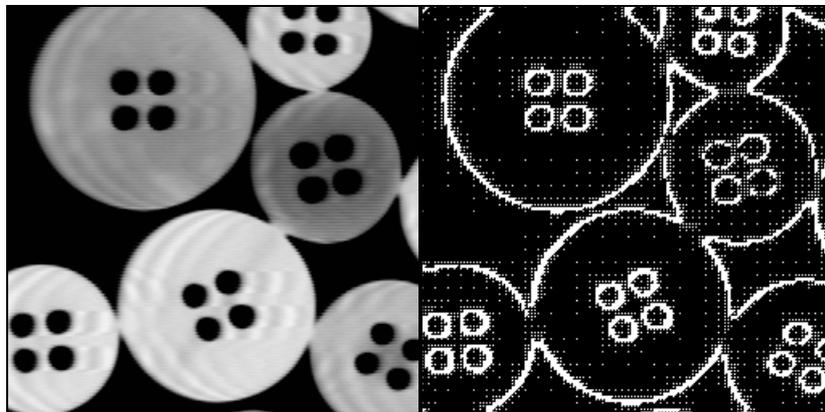


Figura 4.23. Imagen obtenida (derecha) mediante descomposición quad-tree de una imagen filtrada mediante filtrado de mediana (izquierda).

SEGMENTACIÓN MEDIANTE EXTRACCIÓN Y ANÁLISIS DE TEXTURAS

Dentro de las técnicas de segmentación actuales, cobran gran importancia todas aquellas que tratan las texturas que se encuentran en la imagen. Realmente, no hay una única y universal definición de textura, aunque se suele determinar que la medida de texturas es la medición en una imagen de la rugosidad, suavidad y regularidad dentro de ella.

Ehrich y Foith (1978) resumieron el análisis de texturas en tres campos de estudio:

Dada una región con texturas determinar a que tipo pertenece (dentro de un número finito de clases definido)

Dada una textura obtener una descripción o modelo de ella.

Dada una imagen con múltiples texturas, determinar los contornos entre regiones con distintas texturas.

El punto uno se aproxima al reconocimiento de formas. El punto dos se basa en la forma de modelizar las texturas. Y el tercer punto trata de la segmentación de una imagen mediante el análisis de las texturas.

Es claro, que el punto tres no puede hacerse sin el uso de los otros dos (modelizado y clasificación).

Existe gran cantidad de métodos de segmentación basados en el análisis de texturas. Estos se basan en obtener unos descriptores que permitan clasificar claramente las texturas (media, varianza, etc.), para poder extraerlas de la imagen a analizar.

En la bibliografía especializada existen gran cantidad de algoritmos de análisis de texturas.

4.2.4 TÉCNICAS DE CLUSTERIZADO

Las técnicas de clusterizado o agrupación de datos es un método muy utilizado en la clasificación automática de objetos.

En este caso, se trata de definir un conjunto de características asociadas a cada pixel generando un vector multidimensional por cada punto de la pantalla. La técnica de clusterizado se encarga de agrupar dichos vectores según su grado de semejanza entre ellos y así, finalmente se obtienen N grupos de pixels con características semejantes entre ellos.

Según Coleman y Andrews, se puede plantear múltiples parámetros discriminantes de cada pixel. Por ejemplo, para una imagen monocolor se aconsejaba el uso de doce características: luminancia, módulo y fase de un filtrado espacial para la extracción de bordes, las modas de las intensidades luminosas con ventanas de 3×3 , 7×7 y 15×15 , etc.

Para una imagen color, se tomaron características parecidas pero aplicadas a los tres niveles de color por separado: rojo, verde y azul.

Existen múltiples algoritmos de clusterizado cuyo funcionamiento es diferente. Algunos necesitan saber el número de grupos que tienen que clasificar, mientras que otros determinan cuantos grupos de vectores se pueden crear y el centro de gravedad de cada grupo. Por supuesto, estos algoritmos dependen de otros parámetros de entrada que tienen que ser inicialmente ajustados por el humano, ya que los resultados finales pueden ser completamente distintos.

Un tipo de algoritmo de clusterizado bastante avanzado es el **clusterizado substractivo**. Este procedimiento encuentra el número de grupos y los centros de gravedad de estos grupos. El funcionamiento es el siguiente:

Calcula el potencial de cada punto con respecto a los demás. El potencial es un valor que es directamente proporcional al número de puntos vecinos e inversamente proporcional a la distancia de estos. Es decir, si un punto tiene muchos puntos cercanos a él, su potencial será muy elevado.

Se busca el punto de mayor potencial de todos, este será el primer centro elegido (siempre que el potencial sea mayor que un mínimo prefijado).

Una vez elegido el primer centro de clusterizado, se anulan todos los puntos que estén a una distancia prefijada de él.

Se busca el siguiente punto con potencial más elevado, (siempre que supere un umbral predeterminado) y se vuelven a anular los puntos cercanos a él.

Se realiza el punto 4 hasta que los puntos que queden fuera de los grupos tengan un potencial menor que el estipulado.

4.2.5 SEGMENTACIÓN POR MOCIÓN

Un campo de la segmentación en el que se está estudiando mucho, es el de la segmentación de objetos en movimiento. En este caso, la segmentación se aplica a varias imágenes obtenidas en el tiempo, de forma que por comparación se detecta el objeto en movimiento y se extraen sus características.

Este campo, como se puede suponer, es de gran importancia para la robótica y los sistemas de control de calidad.

4.2.6 OTRAS TÉCNICAS DE SEGMENTACIÓN

Aquí se han enumerado las técnicas de segmentación más comunes que se pueden encontrar fácilmente en la bibliografía especializada.

Realmente el número de técnicas es enorme: uso de redes neuronales, técnicas fuzzy, algoritmos específicos para diversas áreas (RWSEG, ANNEAL, etc.), con funciones de contorno, etc. Dado el carácter introductorio de este documento, no se pretende entrar en el desarrollo de todas estas técnicas.

5. ANÁLISIS

Una vez realizado el proceso de segmentación, donde se obtienen los bordes de los objetos bien definidos, el siguiente paso consiste en obtener parámetros que definan las características de cada objeto: forma, textura, color, orientación, etc.

De entre todos los parámetros o descriptores que se puedan obtener, habrá que seleccionar aquellos que tengan las siguientes características:

Ser discriminantes. Es decir, que diferencien lo mejor posible los objetos de una familia o clase con los de las otras.

Ser independientes entre sí. Los descriptores que definan cada objeto no tienen que estar relacionados, de forma que si varía uno los demás no varíen por culpa de éste.

Ser suficientes. Tienen que delimitar de forma suficiente la pertenencia de un objeto a una clase determinada.

Dentro de los descriptores se definen tres categorías básicas: descriptores de frontera, descriptores de región y descriptores adecuados para representar estructuras tridimensionales. En este capítulo se van a tratar los dos primeros grupos.

5.1 REPRESENTACIONES DE LAS FRONTERAS

5.1.1 CÓDIGOS DE CADENA

Los códigos de cadena son usados para definir, a partir de los puntos del contorno de un objeto, las direcciones que van tomando el conjunto de segmentos que forman la frontera del objeto. En la práctica se suelen utilizar el **código 4** o el **código 8**, aunque se pueden utilizar códigos de cadena con más direcciones.

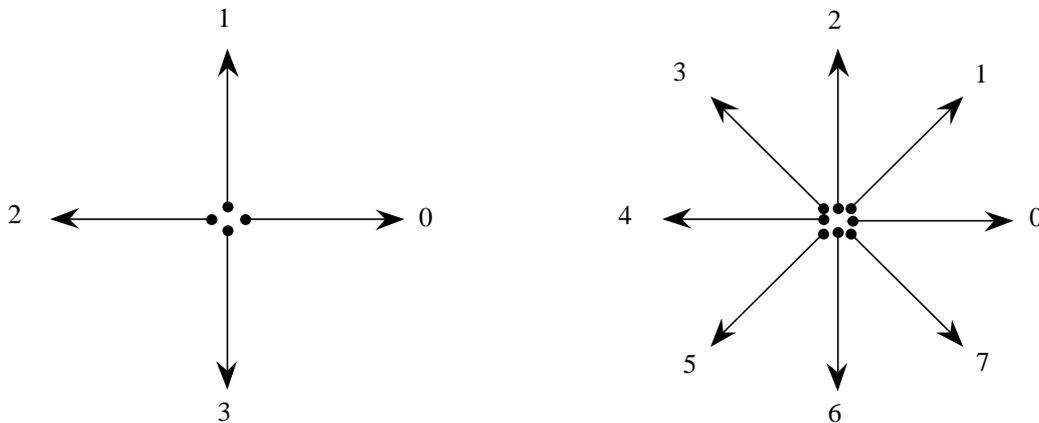


Figura 5.1. Códigos de cadena. Cuatro direcciones (izquierda), ocho direcciones (derecha).

Según la Figura 5.1 el código que se obtendría de la **¡Error! No se encuentra el origen de la referencia.** (según el código 4-direccional), sería: {0, 3, 0, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1, 1, 0, 0, 0, 1}.

NORMALIZACIÓN DEL CÓDIGO DE CADENA

Como se puede observar, el código de cadena depende del punto de comienzo. Para normalizarlo existen diferentes métodos.

Por ejemplo, se puede tratar el código como una secuencia circular donde se va moviendo el punto de comienzo hasta que **el sumatorio de los valores sea mínimo**. Otra forma consistiría en minimizar las diferencias de los valores del código de cadena.

Por otro lado, se puede normalizar en tamaño ajustando el número de los elementos del código de cadena a un valor fijo.

5.1.2 SIGNATURAS

La *signatura* es la representación unidimensional de una frontera o contorno de un objeto y puede ser generada de varias formas, pero, independientemente de cómo sea generada ésta, el objetivo es reducir la frontera a una representación unidimensional, que en principio será más fácil de describir que la frontera original bidimensional.

Las *signaturas* más conocidas son:

Representar la distancia del centro hasta cada punto del contorno en función del ángulo. Se puede normalizar el tamaño máximo a 1.

Recorrer la frontera y representar el ángulo de la tangente al contorno en cada punto.

La *función de densidad de pendiente*, donde se representa un histograma de los valores del ángulo tangente en los puntos del contorno. Este histograma define claramente la forma de un objeto (líneas, curvas, etc.).

5.1.3 APROXIMACIONES POLIGONALES

El objetivo de la aproximación poligonal es capturar la esencia de la forma del contorno. Consiste en adaptar una forma poligonal a la del objeto.

Existen multitud de métodos más o menos complejos de aproximación de una forma poligonal al contorno de un objeto. Algunos obtienen una ecuación polinómica del objeto mediante métodos matemáticos mientras que otros usan aproximaciones geométricas.

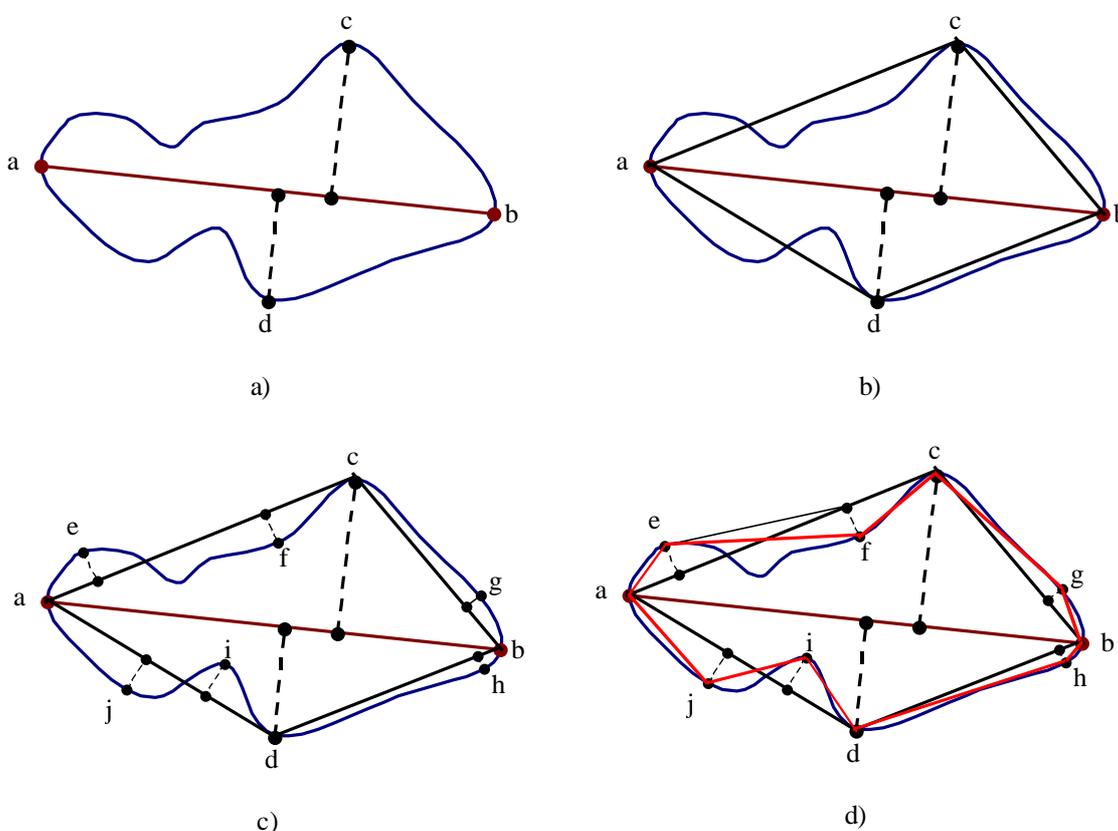


Figura 5.2. Obtención de la forma poligonal de un contorno.

Un método muy conocido de aproximación geométrica, consiste en:

Trazar la recta entre los dos puntos más distantes (puntos a y b) del contorno y obtener los puntos más alejados a ésta (puntos c y d).

Trazar las rectas entre los puntos obtenidos (Figura 5.2.b)

Realizar el paso 1 con las nuevas rectas obtenidas (Figura 5.2.c y Figura 5.2.d). Repetir hasta que se cumpla un cierto criterio: distancia máxima mayor que un umbral, número fijo de iteraciones, número de polígonos, etc.

5.2 REPRESENTACIONES DE LAS REGIONES

La forma de un objeto puede representarse directamente por la región que ocupa. Por ejemplo, la matriz binaria:

$$u(m, n) = \begin{cases} 1 & \text{si } (m, n) \in \mathcal{R} \\ 0 & \text{en caso contrario} \end{cases}$$

es una representación de la región \mathcal{R} . Obviamente la representación por fronteras es más eficiente que este procedimiento.

5.2.1 CÓDIGOS DE LONGITUD VARIABLE

Cualquier región o imagen binaria puede ser vista como una sucesión en la que se alternan hileras de ceros y unos. Los códigos de longitud variable representan estas hileras. Para una imagen que es escaneada por filas hasta llegar a la región de interés, el código consiste en la dirección de comienzo de cada hilera seguida del número de unos o ceros que la integran. Un ejemplo se encuentra en la Figura 5.3.

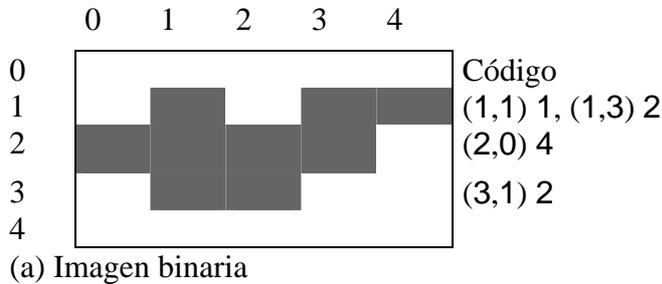
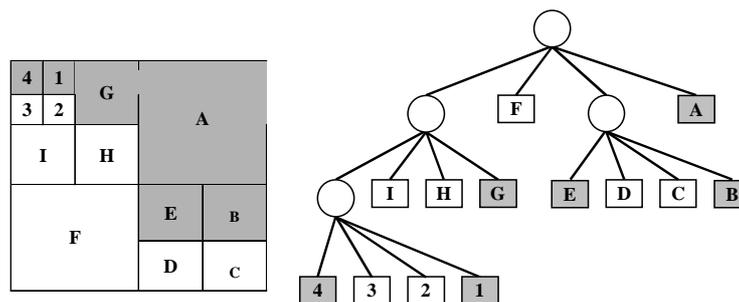


Figura 5.3. Codificaciones de una región. (a) Imagen binaria y su codificación.

5.2.2 QUAD-TREES

En este método, la región dada se incluye en un área rectangular conveniente. Este área es dividido en cuatro cuadrantes y se examina cada uno de ellos para ver si todos sus pixels pertenecen a la misma región o no. Si tiene pertenecientes y no pertenecientes a la región, se sigue subdividiendo. Una estructura de árbol se genera hasta que cada subcuadrante sólo contiene pixels de un tipo. Supóngase que se utiliza el 1 para la región y el 0 para la no región, entonces el árbol puede codificarse de forma única como una hilera de ceros, unos y 'g' para denotar no región, región y subdivisión necesaria (ver Figura 5.4).



Código: g 1 g1001 0 g100g1001

(a) Cuadrantes (b) Creando el quad-tree

Figura 5.4. (a) Región, (b) Quad-tree y su representación.

Es claro que esta representación está más comprimida que si se utilizan códigos de longitud variable, sin embargo, el cálculo de descriptores como perímetro, área, etc, es más complejo.

5.2.3 PROYECCIONES

Una forma bidimensional o región \mathcal{R} puede ser representada también por sus proyecciones. Una proyección $g(s, \theta)$ es simplemente la suma de las longitudes de las hileras a lo largo de una línea recta orientada a un ángulo θ y a una distancia s . En este sentido una proyección es un histograma que proporciona el número de pixels que se proyectan en una determinada posición a una distancia s en la línea de orientación θ (ver Figura 5.5).

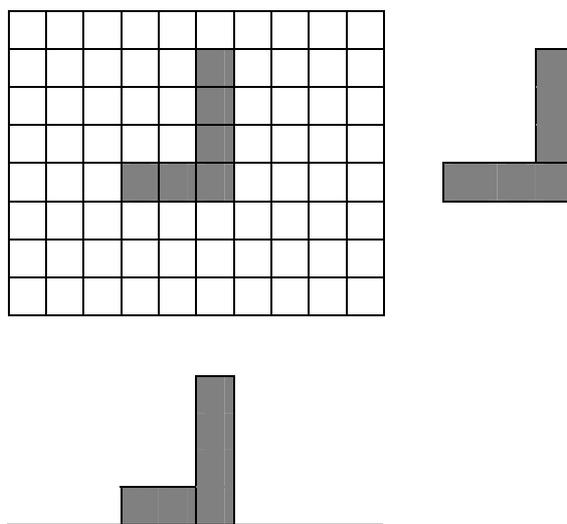


Figura 5.5. Una imagen y dos de sus proyecciones.

Es obvio que variando el ángulo se obtiene mucha información sobre la forma del objeto. Además, las proyecciones pueden servir para intentar reconstruir el objeto a partir de estas. Este es un aspecto muy importante de aplicación de la teoría de proyecciones. En particular la aplicación es básica en el campo de la Medicina con la utilización de imágenes PET y SPECT.

5.3 DESCRIPTORES DE FRONTERA

Son parámetros que se obtienen del estudio del contorno de los objetos.

5.3.1 LONGITUD

La *longitud* de un contorno es uno de los descriptores más sencillos. Si se cuenta el número de pixels en el contorno se obtendrá una aproximación de su longitud. Si la frontera ha sido codificada usando cadenas de códigos, el número de componentes verticales más el de horizontales más el de diagonales multiplicados por $\sqrt{2}$ da la longitud exacta.

5.3.2 DIÁMETRO

El *diámetro* de una frontera B se define como:

$$Diam(B) = \max_{i,j} [D(p_i, p_j)] \quad (5.1)$$

donde D es una medida de distancia y p_i y p_j son puntos de la frontera. Tanto el diámetro como la línea que une los dos puntos que definen el diámetro (llamada *eje mayor* de la frontera) son descriptores útiles.

5.3.3 CURVATURA

La *curvatura* se define como la razón de cambio de la pendiente. En general, obtener la curvatura de una forma fiable es difícil ya que el ruido y otros factores proporcionarán una curvatura con demasiadas oscilaciones locales. Sin embargo, si se usa la diferencia de las pendientes de segmentos de fronteras adyacentes (que se representarán como líneas rectas) como un descriptor de la curvatura en el punto de intersección de los segmentos puede ser bastante útil.

Cuando una frontera es recorrida en la dirección de las agujas del reloj, un vértice p se dice parte de un segmento convexo si el cambio en la pendiente en p es no negativo; en caso contrario se dice cóncavo.

5.3.4 DESCRIPTORES DE FOURIER

La transformada de Fourier se utiliza en el **análisis frecuencial** de señales discretas. En este caso, la utilidad de aplicarla para la descripción de objetos consiste en que, generalmente, con los primeros valores que devuelve esta transformada (correspondientes a las bajas frecuencias) **permite diferenciar objetos distintos de una forma rápida y precisa.**

Consiste en introducir como valor complejo cada punto del contorno y realizarle la transformada FFT (Figura 5.6).

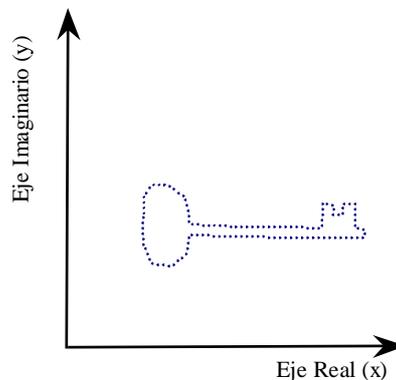


Figura 5.6. Representación de un contorno en el espacio de los números complejos.

5.4 DESCRIPTORES DE REGIÓN

Los descriptores de región se pueden obtener del análisis del propio contorno del objeto o por características internas del mismo.

A continuación se muestran las más utilizadas.

5.4.1 MEDIDA DE ÁREA

Dentro de un objeto se pueden realizar diferentes medidas de áreas: del objeto, de sus huecos, etc.

ÁREA DEL OBJETO

Para realizar la medida del área de un objeto, se utiliza el vector que define su contorno. El área se va a definir como el número de pixels que están dentro del contorno.

Uno de los métodos más rápidos, consiste en contar el número de pixels que hay en cada fila de puntos del objeto.

			X	1	1	1	X			=	5
		X	1	1	1	1	X			=	6
	X	1	1	1	1	1	1	X		=	8
X	1	1	1	1	1	1	1	1	X	=	10
X	1	1	1	1	1	1	1	1	X	=	10
X	1	1	1	1	1	1	1	1	X	=	10
X	1	1	1	1	1	1	1	1	X	=	10
	X	1	1	1	1	1	1	1	X	=	9
	X	1	1	1	1	1	1	X		=	8
		X	1	1	1	1	X			=	6
			X	1	1	1	X			=	5

Figura 5.7. Contorno de un agujero, con X se marcan para cada fila puntos con columnas mínima y máxima.

De esta forma el cálculo del área es muy sencillo:

$$\text{Área} = \sum_{h=\min(\text{filas})}^{\max(\text{filas})} \max(\text{columna}(h)) - \min(\text{columna}(h)) \quad (5.2)$$

ÁREA DE LOS HUECOS

Otro parámetro importante, es el cálculo del área de los huecos que estén dentro del objeto. Si éste tiene varios, se devuelve la suma de todas las áreas.

El proceso de cálculo es igual que el anterior.

ÁREA MEDIA DE HUECOS

Se define como:

$$\text{Area_media_huecos} = \text{Área_huecos} / \text{Num_huecos} \quad (5.3)$$

ÁREA DE PERFORACIÓN

Viene definida como:

$$\text{Area_de_perforación} = \text{Área_huecos} / \text{Área_objeto} \quad (5.4)$$

Da una idea, de la densidad de pixels que pertenecen al objeto.

ÁREA CONVEXA

El valor de vuelta especifica una aproximación del área convexa de un objeto. Se obtiene calculando los diversos diámetros de Feret con diferentes ángulos. Cuantos más ángulos se utilicen, la aproximación será mayor. Sin embargo, esto supone mayor tiempo de cálculo. Los huecos no son analizados.

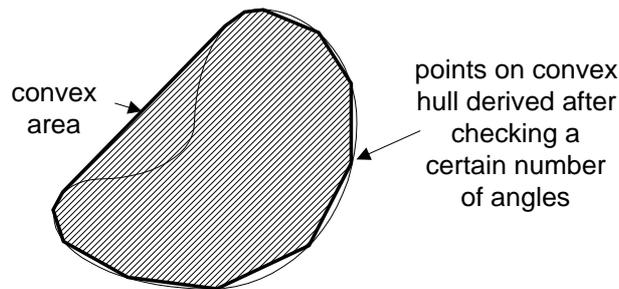


Figura 5.8. Obtención del área convexa de un objeto.

5.4.2 CENTRO DE GRAVEDAD

El centro de gravedad o mejor llamado *Centroide*, se puede calcular de dos formas: como la media de todas las coordenadas de los puntos del contorno, o como la media de todas las coordenadas de los puntos internos del objeto. Este segundo método, aunque mucho más lento, es más preciso. Dado un objeto formado por un conjunto de puntos etiquetados con un valor determinado, el cálculo de la coordenada X_C del centro de gravedad (el de la coordenada Y_C será de la misma forma será:

$$X_C = \frac{m_{10}}{m_{00}}$$

$$m_{10} = \sum_{y=0}^{R-1} \sum_{x=0}^{C-1} x * p(x, y) \quad m_{00} = \sum_{y=0}^{R-1} \sum_{x=0}^{C-1} p(x, y)$$

where

X_C = x - position of the center of gravity

x = the pixels x - coordinate

p(x,y) = gray value of pixel; set to 1 for binary moments

R = number of rows

C = number of columns

Figura 5.9. Método general para el cálculo de la coordenada X_C del centro de gravedad.

5.4.3 CAJA DEL CONTORNO

Corresponde al rectángulo que se ajusta al contorno. Está formado por las coordenadas $XMIN$, $YMIN$, $XMAX$ e $YMAX$.

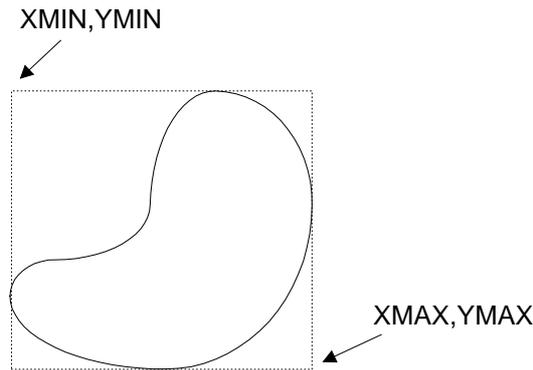


Figura 5.10. Cuadrado que se ajusta al contorno del objeto.

5.4.4 MOMENTOS CENTRALES

Los momentos se dividen en: momentos ordinarios y momentos centrales. Los primeros se calculan con coordenadas

MOMENTOS ORDINARIOS

Se calculan con la relación al sistema de coordenadas relativas al origen de la imagen. Se obtienen de la siguiente manera:

$$m_{ij} = \sum_{y=0}^{R-1} \sum_{x=0}^{C-1} x^i * y^j * p(x, y)$$

where

m_{ij} = moment

x^i = the pixels x - coordinate

y^j = the pixels y - coordinate

$p(x, y)$ = gray value of pixel; set to 1 for binary moments

R = number of rows

C = number of columns

$i, j = 0, 1, 2, \dots$

whereas $(i + j)$ define the order of the moment

Figura 5.11. Método para el cálculo de los momentos ordinarios.

MOMENTOS CENTRALES

Se usan las coordenadas relativas al centro de gravedad del objeto.

$$\mu_{ij} = \sum_{y=0}^{R-1} \sum_{x=0}^{C-1} (x - x_c)^i * (y - y_c)^j * p(x, y)$$

where

μ_{ij} = central moment

x^i = the pixels x - coordinate

y^j = the pixels y - coordinate

x_c = x - coordinate of center of gravity

y_c = y - coordinate of center of gravity

$p(x, y)$ = gray value of pixel; set to 1 for binary moments

R = number of rows

C = number of columns

$i, j = 0, 1, 2, \dots$

whereas $(i + j)$ define the order of the moment

Figura 5.12. Método para el cálculo de los momentos ordinarios.

5.4.5 CUERDAS

Se definen “cuerdas” como las líneas de “scan” horizontales de una imagen.

NÚMERO DE CUERDAS

Corresponde al número de filas de pixels que forman el objeto.

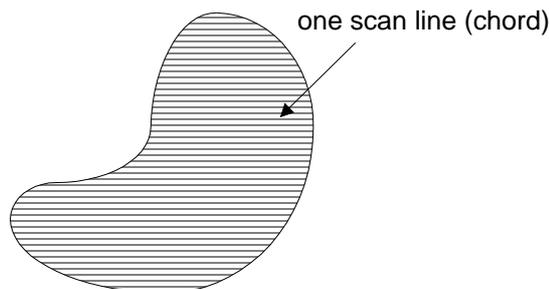


Figura 5.13. Cuerdas que forman un objeto.

SUMA DE LAS LONGITUDES DE LA LÍNEAS

Corresponde a la suma, en pixels, de las longitudes de todas las cuerdas que componen el objeto.

LONGITUD MEDIA DE LA CUERDA

Es la media de las longitudes de las cuerdas. El cálculo se realiza de la forma siguiente:

$$\text{MeanChordLength} = \frac{\text{ChordSum}}{\text{ChordNumber}}$$

where
 ChordSum = length of all scan lines in pixels
 ChordNumber = number of scanlines

Figura 5.14. Cálculo de la longitud media de las cuerdas.

5.4.6 PERÍMETRO

Existen dos formas de obtener el perímetro de un objeto.

PERÍMETRO CONVEXO

Se obtiene mediante el cálculo de diferentes diámetros de Feret con distintos ángulos. El resultado obtenido es aproximado.

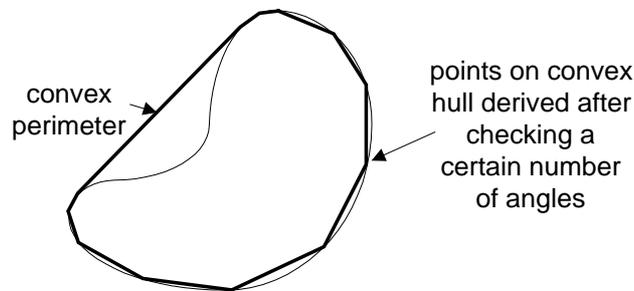


Figura 5.15. Perímetro convexo.

PERÍMETRO NORMAL

Se determina sumando los pixels del contorno y multiplicándolos por una factor. Los puntos horizontales y verticales tendrán un factor de 1, mientras que los diagonales su factor es de $\sqrt{1^2 + 1^2} = \sqrt{2}$ correspondiente a la longitud de la diagonal del pixel.

De esta forma, el cálculo del perímetro utilizando el contorno se obtendrá con la siguiente fórmula:

$$p = n_x s_x * n_y s_y * n_d \sqrt{s_x^2 + s_y^2}$$

where
 p = perimeter of the object
 n_x = number of horizontal steps
 n_y = number of vertical steps
 n_d = number of diagonal steps
 s_x = scale factor in x - direction
 s_y = scale factor in y - direction

Figura 5.16. Perímetro normal del objeto.

5.4.7 DIÁMETROS DE FERET

Los diámetros de Feret son muy utilizados en la medida geométrica de objetos. Existen varios tipos.

DIÁMETRO DE FERET GENERAL

El diámetro de Feret consiste en determinar la máxima longitud de la cuerda para un ángulo en sentido antihorario con el eje x y que se define entre 0 y 180° .

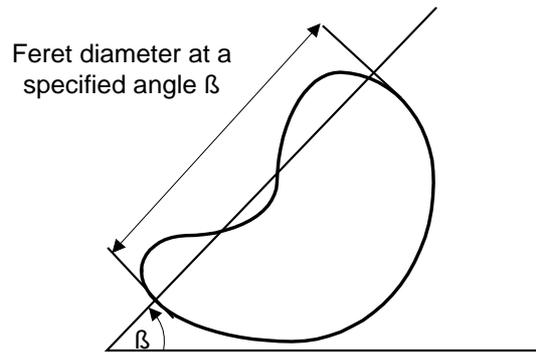


Figura 5.17. Diámetro de Feret.

DIÁMETRO FERET HORIZONTAL Y VERTICAL

Corresponde al diámetro de la proyección horizontal y vertical del objeto.

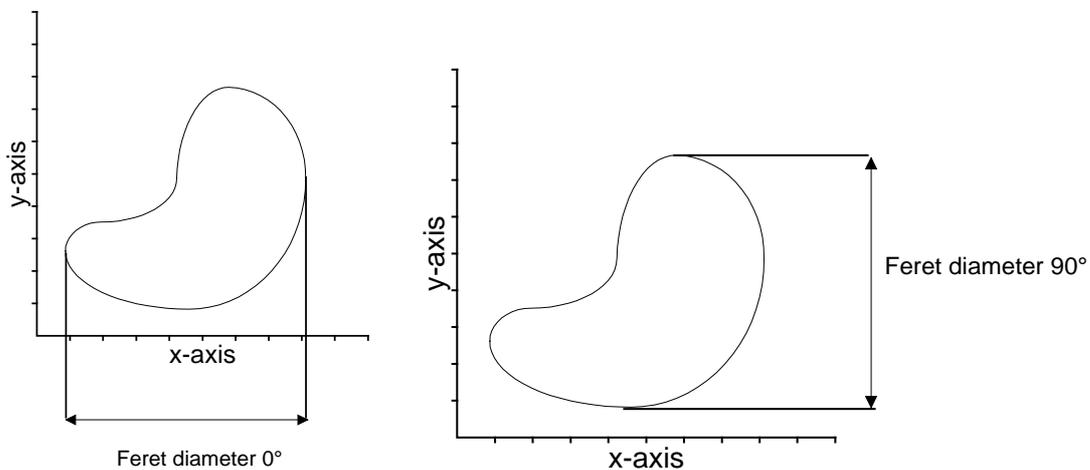


Figura 5.18. Diámetros de Feret horizontal y vertical.

ÁNGULO Y DIÁMETRO DE FERET MÁXIMOS Y MÍNIMOS

Corresponde al valor del ángulo cuyo diámetro de Feret es máximo y mínimo respectivamente.

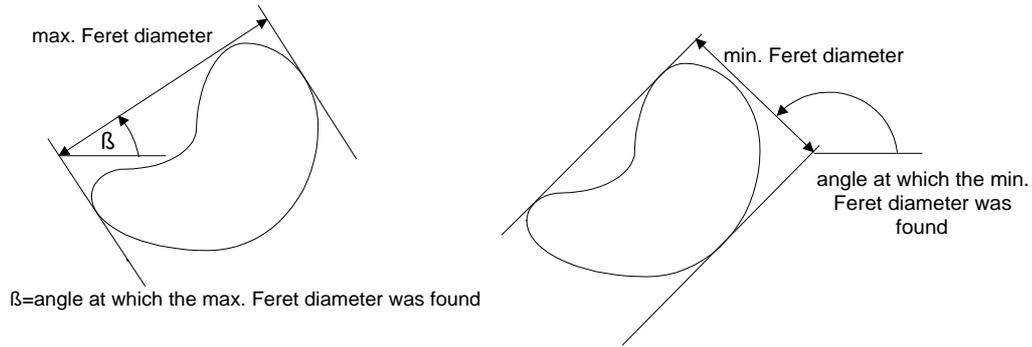


Figura 5.19. Ángulos cuyo diámetro de Feret es máximo (izquierda) y mínimo (derecha).

LONGITUD MEDIA DE LOS DIÁMETROS DE FERET

Es igual a la media de todas las longitudes de los diámetros Feret obtenidos con diferentes ángulos.

RELACIÓN DE ASPECTO DE FERET

Consiste en relacionar los diámetros de Feret máximo y mínimo de un objeto.

$$Relación_de_Aspecto = \frac{Diámetro_Feret_máximo}{Diámetro_Feret_mínimo} \tag{5.5}$$

5.4.8 PUNTOS SIGNIFICATIVOS DEL CONTORNO

Dentro del contorno se pueden definir multitud de puntos significativos:

Puntos más alto, bajo, a la izquierda y a la derecha.

Primer punto que obtiene al buscar de izquierda-derecha y de arriba-abajo.

Puntos extremos: superior-izquierda, superior-derecha, derecha-superior, derecha-inferior, inferior-derecha, inferior-izquierda, izquierda-inferior e izquierda-superior.

5.4.9 EJES

Corresponden a los ejes que atraviesan el centro de gravedad del objeto. Generalmente se buscan los ejes mayores y menores del objeto.

EJE MAYOR Y MENOR

Los ejes más significativos de un objeto son los de mayor y menor longitud, y los horizontales y verticales.

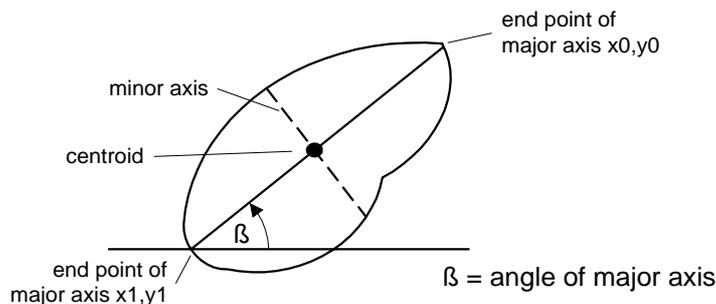


Figura 5.20. Ejes máximo y mínimo de un objeto.

En la Figura 5.20 se pueden observar los ejes máximo y mínimo. Los parámetros más importantes que definen cada eje son: coordenadas de los puntos extremos de los ejes, el ángulo respecto al eje y longitud del mismo.

5.4.10 ORIENTACIÓN

El parámetro siguiente sirve para determinar la orientación del objeto.

ÁNGULO DE ORIENTACIÓN PRINCIPAL DEL OBJETO

Se obtiene con la siguiente fórmula:

$$\beta = \arctan \frac{\lambda_1 - v_{xx}}{v_{xy}}$$

where
 λ_1 = Eigenvalue of the covariance matrix
 v_{xx}, v_{xy} = elements of the covariance matrix

The covariance matrix is defined as follows:

$$C = \begin{pmatrix} v_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{pmatrix}$$

where

$$v_{xx} = \frac{\sum_x \sum_y (x - x_c)^2}{A}, v_{yy} = \frac{\sum_x \sum_y (y - y_c)^2}{A}$$

$$v_{xy} = v_{yx} = \frac{\sum_x \sum_y (x - x_c)(y - y_c)}{A}$$

where
 A = area of the object
 x_c, y_c = center of gravity x- and y- coordinate

Figura 5.21. Cálculo de la orientación de un objeto.

ÁNGULO PERPENDICULAR A LA ORIENTACIÓN PRINCIPAL DEL OBJETO

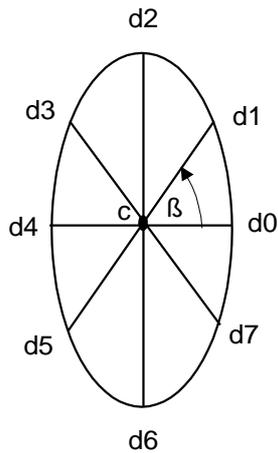
Consiste en obtener el ángulo perpendicular del anterior.

5.4.11 MEDIDAS DE DISTANCIA POLAR

Se utilizan las medidas con el cálculo polar respecto del centro de gravedad del objeto.

DISTANCIA POLAR

Se calcula la distancia máxima desde el centro de gravedad al contorno con un ángulo determinado.



d0..d7 : distances from centroid (c) at equal spaced angles (β)

Figura 5.22. Cálculo de la distancia polar para diversos ángulos.

DISTANCIA POLAR MÁXIMA Y MÍNIMA

Consiste en determinar las distancias máximas y mínimas, el ángulo de cada una de ellas y las coordenadas correspondientes a cada punto del contorno.

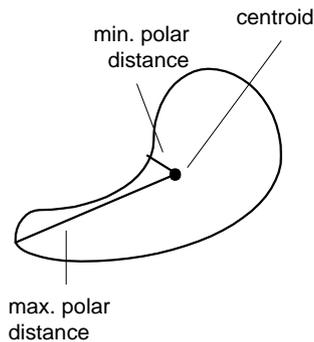


Figura 5.23. Cálculo de las distancias polares máxima y mínima.

5.4.12 RUGOSIDAD

Permite determinar la rugosidad de un objeto. Un objeto liso tendrá rugosidad 1.0.

$$Rugosidad = \frac{Perímetro}{Perímetro_Convexo} \quad (5.6)$$

5.4.13 REDONDEZ

Sirve para determinar la redondez de objetos redondos. Un objeto redondo perfecto, tendrá una redondez de 1.0, mientras que los objetos rectos y alargados se acercarán a cero.

Se obtiene de la relación entre el área calculada y el área de un círculo:

$$Redondez = \frac{4 \cdot \pi \cdot Area_Objeto}{Perímetro^2} \quad (5.7)$$

$$Redondez = \frac{Perímetro}{2 \cdot \pi \cdot Radio_Mínimo} \quad (5.8)$$

$$Redondez = \frac{\pi \cdot Radio_Mínimo^2}{Area_Objeto} \quad (5.9)$$

donde el *Radio_Mínimo* es el equivalente a la distancia polar mínima.

5.4.14 NÚMERO DE HUECOS

Corresponde al número de huecos que hay dentro de un objeto.

5.4.15 NÚMERO DE EULER

Es igual a:

$$Número_Euler = 1 - Número_huecos \quad (5.10)$$

5.5 MEDIDAS DENSITOMÉTRICAS

Existen múltiples parámetros que permiten definir un objeto, según la densidad de los puntos del mismo.

5.5.1 FUNCIÓN DE AUTOCORRELACIÓN

La función de autocorrelación se calcula con la matriz de concurrencia.

$$AutoCorrelation = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} ij * C_{\delta}(i, j)$$

where
 $C_{\delta}(i, j)$ = cooccurrence matrix
 i, j = gray values at position $\delta(dx, dy)$
 L = number of distinct gray values

Figura 5.24. Cálculo de la autocorrelación.

5.5.2 ENERGÍA

La energía (también llamada “momento segundo angular”) de un objeto se calcula utilizando la matriz de concurrencia de la siguiente forma:

$$Energy_{\delta} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} C_{\delta}(i, j)^2$$

where
 $C_{\delta}(i, j)$ = cooccurrence matrix
 L = number of distinct gray values
 i, j = gray values at position $\delta(dx, dy)$

Figura 5.25. Cálculo de la energía de un objeto.

5.5.3 PARÁMETROS ÓPTICOS

DENSIDAD ÓPTICA

Se define como:

$$\text{IGV} = \text{Area} * \frac{\sum_{z=1}^{z_{\max}} \log_{10}\left(\frac{z_{\max}}{z}\right) * H(z)}{N_{0 \notin N}}$$

where
 IGV = Integrated Gray Value
 Area = area of object (number of foreground pixel)
 $N_{0 \notin N}$ = number of pixel of the object without pixel having a gray value of 0
 $H(z)$ = absolute histogram distribution, z = gray value

Figura 5.26. Cálculo de la densidad óptica de un objeto.

DENSIDAD ÓPTICA MEDIA

Consiste en obtener la intensidad media de un objeto.

$$\text{MIGV} = \frac{\text{IGV}}{\text{Area}}$$

where
 MIGV = Mean Integrated Gray Value
 IGV = Integrated Gray Value
 Area = area of object (number of foreground pixel)

Figura 5.27. Cálculo de la densidad óptica media de un objeto

5.5.4 PARÁMETROS ESTADÍSTICOS

Se muestran otros parámetros típicos:

SUMA Y SUMA CUADRADA

Consiste en obtener la suma de los valores de los pixels del objeto.

Otro parámetro, muy común que se suele obtener es la suma cuadrada de los valores de luminancia del objeto.

MÍNIMO Y MÁXIMO

Consiste en obtener el valor de intensidad máximo y mínimo de un objeto.

VALOR MEDIO

Se realiza la media de los valores de intensidad del objeto.

$$\bar{p} = \frac{\sum_{i=0}^{N-1} p(i)}{N}$$

where
 \bar{p} = mean gray value
 N = number of pixels
 p = pixel value

Figura 5.28. Cálculo de la densidad óptica media de un objeto.

5.5.5 DESCRIPCIÓN DE TEXTURAS

La descripción simple de texturas, está basada en el histograma del objeto $p_f(f_k)$, siendo f_k , $k=1, \dots, N$, los distintos valores de intensidad de éste.

Algunos de los momentos más utilizados en la descripción de texturas son:

MEDIA

Determina una media estimada de los niveles de grises en la región analizada.

$$\mu = \sum_{k=1}^N f_k \cdot p_f(f_k) \quad (5.11)$$

VARIANZA

La varianza determina la dispersión de intensidad en el objeto a estudiar.

$$\sigma^2 = \sum_{k=1}^N (f_k - \mu)^2 \cdot p_f(f_k) \quad (5.12)$$

ONDULACIÓN (SKEWNESS)

Sirve para analizar la simetría del histograma.

$$\mu_3 = \frac{1}{\sigma^3} \cdot \sum_{k=1}^N (f_k - \mu)^3 \cdot p_f(f_k) \quad (5.13)$$

KURTOSIS

Kurtosis permite analizar los extremos del histograma. Histogramas con largos extremos corresponden a regiones en forma de púas.

$$\mu_4 = \frac{1}{4} \cdot \sum_{k=1}^N (f_k - \mu)^4 \cdot p_f(f_k) - 3 \quad (5.14)$$

ENTROPÍA

La entropía de una imagen se define como:

$$H = - \sum_{k=1}^N p_f(f_k) \cdot \ln p_f(f_k) \quad (5.15)$$

5.6 MEDIDAS DE LÍNEAS Y PUNTOS

De los puntos anteriores, se puede extraer la medición en una dimensión para líneas y puntos.

5.6.1 MEDIDAS DE LÍNEAS GEOMÉTRICAS

Dadas dos líneas se pueden obtener de ellas, múltiples parámetros, como por ejemplo: longitud de las mismas, ángulo que forman con una línea referencia, etc.

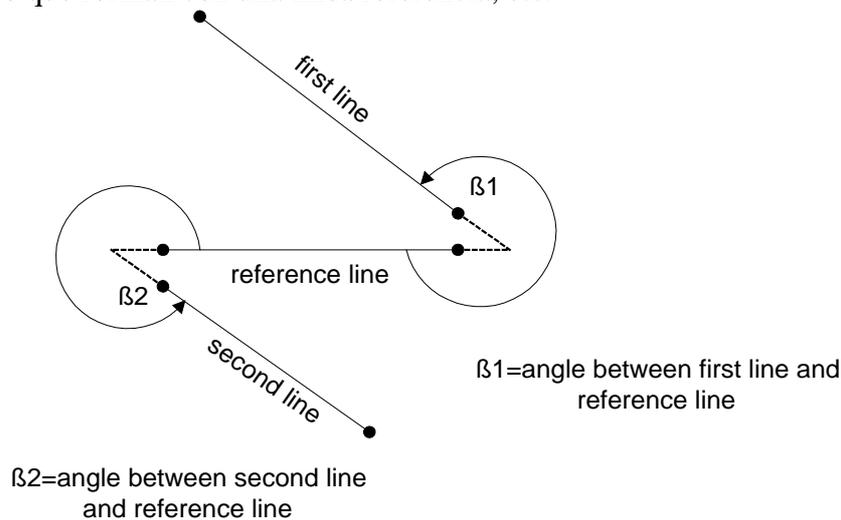


Figura 5.29. Obtención del ángulo entre dos líneas y una de referencia.

MEDIDAS DE LÍNEA DENSITOMÉTRICAS

Consiste en trasladar las medidas densitométricas vistas en el apartado anterior, pero en una dimensión: media, máximo, mínimo, etc.; de los valores de luminancia de los pixels que forman la línea.

MEDIDAS DE ÁNGULOS ENTRE PUNTOS

Se pueden calcular según el número de puntos que se traten: dos, tres, cuatro, etc.

Con dos puntos se medirá el ángulo que forma la línea que los une con la línea horizontal o vertical.

Con tres, se calculará el ángulo que forman las rectas formadas por estos tres puntos.

Con cuatro, se emparejarán de dos en dos y se calculará el ángulo entre las rectas formadas por las parejas de puntos.

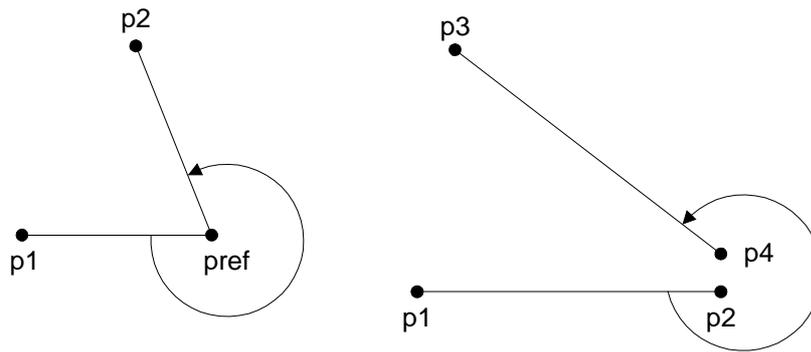


Figura 5.30. Obtención del ángulo entre tres puntos (izquierda) y cuatro puntos (derecha) respectivamente.

MEDIDAS DENSITOMÉTRICAS DE PUNTOS

Lo anteriormente aplicado a superficies y líneas, se puede aplicar a un conjunto de puntos debidamente seleccionados

