

UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA

Departamento de Inteligencia Artificial

TESIS DOCTORAL

CONTRIBUCIÓN AL RECONOCIMIENTO
DE OBJETOS 2D MEDIANTE DETECCIÓN
DE BORDES EN IMÁGENES EN COLOR

Nicolás Luis Fernández García

Córdoba, mayo de 2.002.

UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA

Departamento de Inteligencia Artificial

TESIS DOCTORAL

CONTRIBUCIÓN AL RECONOCIMIENTO
DE OBJETOS 2D MEDIANTE DETECCIÓN
DE BORDES EN IMÁGENES EN COLOR

Nicolás Luis Fernández García

*Directores: Rafael Medina Carnicer
Ángel Carmona Poyato*

Tutor: Darío Maravall Gómez Allende

Córdoba, mayo de 2.002.

Resumen

El presente trabajo estudia las características esenciales de las fases de una metodología de reconocimiento de objetos bidimensionales basada en la detección de bordes en imágenes en color y en el análisis de la curvatura de los puntos de los contornos de los objetos. Se ha incluido una revisión del “estado del arte” de cada una de las fases y se han propuesto nuevos algoritmos.

Se ha realizado un estudio sobre la influencia, el origen y los métodos de supresión o atenuación del ruido en imágenes digitales.

Se han descrito los tipos de bordes y las principales propuestas sobre algoritmos de detección de bordes, tanto en imágenes monocromáticas como en color.

Se ha diseñado una metodología para comparar medidas de evaluación de algoritmos de detección de bordes. Las pruebas realizadas han permitido determinar una medida de evaluación adecuada para comparar algoritmos de detección de bordes. La medida seleccionada ha sido utilizada para comparar diferentes algoritmos de detección de bordes en imágenes en color y elegir el más eficiente.

Se han estudiado las características del adelgazamiento de bordes, se han descrito algunos de los algoritmos más importantes y se han propuesto dos variantes.

Se ha diseñado un método que permite extraer el contorno exterior de un objeto. Para el desarrollo de dicho método, se han propuesto dos algoritmos de identificación de los puntos del contorno exterior y un algoritmo de enlace que permite el cierre de un contorno.

Se han revisado los sistemas de representación de contornos y los métodos utilizados para extraer los descriptores de un contorno. Se ha desarrollado un sistema de representación de contornos invariante a rotaciones y traslaciones que ha sido utilizado para obtener los puntos críticos o relevantes de los contornos mediante el análisis de los puntos de curvatura extrema.

Se han propuesto dos nuevas funciones de comparación de contornos, basadas en los puntos críticos, que son simétricas y normalizadas.

En resumen, se ha desarrollado una metodología de reconocimiento de objetos bidimensionales basada en el uso de un algoritmo de detección de bordes en imágenes en color y en el análisis de la curvatura de los puntos de los contornos.

Abstract

The present work studies the main characteristics of the stages of bidimensional objects recognition methodology based on edge detection in color images and objects contour points curvature analysis. A revision of the “state of art” of every stage has been included and new algorithms have been proposed.

The influence, origin and the methods to eliminate or attenuate the noise in digital images have been studied.

The types of edges and the main proposals about edge detection, both in monochrome and color images, have been described.

A methodology to compare evaluation measures of edge detection algorithms has been designed. The proofs have determined the evaluation measure more appropriate to compare edge detection algorithms. The selected measure have been used to compare different edge detection algorithms in color images and choose the most efficient one.

The features of edge thinning have been studied, the main algorithms have been described and two versions have been proposed.

A method to extract the object outer contour have been designed. In order to develop this method, two algorithms for identification of outer contour points and a link algorithm, which allows to close the contour, have been proposed.

The contour representation systems and the methods used to extract the contour descriptors have been revised. A contour representation system, which is invariant to rotations and translations, have been developed. This system has been used to obtain the relevant or critical points of the contours by means of extreme curvature points analysis.

Two new comparison functions of contours have been proposed, based on critical points, which are symmetrical and normalized.

In short, a bidimensional objects recognition methodology based on edge detection algorithm in color images and contour points curvature analysis has been developed.

A mis padres.

Agradecimientos

A D. Rafael Medina Carnicer y D. Ángel Carmona Poyato, Catedráticos de Escuela Universitaria del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba, por aceptar la dirección del presente trabajo, su esfuerzo, estímulo y amistad.

A D. Darío Maravall Gómez Allende, Catedrático de Universidad del Departamento de Inteligencia Artificial de la Universidad Politécnica de Madrid, porque sin su colaboración no hubiera sido posible el desarrollo del presente trabajo.

A D. Francisco José Madrid Cuevas, Profesor Titular de Escuela Universitaria del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba, y a D. Manuel Prieto Villegas, Becario del Servicio Informática de la Universidad de Córdoba, por su disponibilidad para superar cualquier dificultad relacionada o no con el procesamiento digital de imágenes.

A D. Manuel Rafael Cosano Jurado, D. Juan Manuel Rodríguez Muñoz, D. Antonio Jesús Cáceres Galiot, D. David Puentes Rojas, D^a. María Del Carmen Gallego Ayllón y D. Antonio José Ravé Gámez, Diplomados en Informática (Especialidad de Sistemas Lógicos) en la Escuela Politécnica Superior de la Universidad de Córdoba, a los que tuve el honor de dirigir sus proyectos de fin de carrera relacionados con temas de detección de bordes en color y adelgazamiento de bordes, porque gracias a su esfuerzo facilitaron que el presente trabajo diera sus primeros pasos.

A D. Juan Antonio Romero Del Castillo y D. Sebastián Ventura Soto, Profesores Titulares de Escuela Universitaria del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba, y a D. Luis Meléndez Aganzo, miembro del Servicio de Informática de la Universidad de Córdoba, por su disponibilidad y ayuda prestada para el desarrollo del presente trabajo.

A los profesores de la Universidad de Córdoba D. Eduardo Gutiérrez de Ravé Agüera, Catedrático de Escuela Universitaria del Departamento de Ingeniería Gráfica e Ingeniería y Sistemas de Información Cartográfica, y D^a. Manuela Luisa Jodral Villarejo, Profesora Titular del Departamento de Bromatología y Tecnología de los Alimentos, y a D^a. Salud Serrano Jiménez, Becaria del Departamento de Bromatología y Tecnología de los Alimentos, por la colaboración prestada al suministrar muchas de las imágenes utilizadas en el presente trabajo y por su estímulo.

Al Servicio de Obtención del Documento de la Biblioteca de la Universidad de Córdoba, en general, y a D^a. Catalina Guzmán Pérez y D^a. María Jesús Morillo Calero, en particular, por facilitar el acceso a muchos de los textos citados en el presente trabajo.

A numerosos miembros de la Universidad de Córdoba, en general, y del Departamento de Informática y Análisis Numérico, en particular, por el interés mostrado en el desarrollo del presente trabajo.

Y, por último, a mi numerosa familia y mis amigos, por el continuo estímulo para que el presente trabajo pudiera ser terminado.

*“Poco se puede esperar de quien sólo se esfuerza cuando tiene la
certidumbre de que va a ser recompensado”.*

José Ortega y Gasset.

“El hombre que sabe no vale tanto como el que ama”.

Proverbio chino.

Índice General

I	Introducción	1
1	Introducción	3
1.1	Procesamiento digital de imágenes	3
1.2	Reconocimiento de objetos	6
1.3	Objetivos	10
1.4	Organización del documento	11
II	Revisión de antecedentes	13
2	Ruido	15
2.1	Introducción	15
2.2	Ruido provocado por la adquisición de la imagen	16
2.3	Ruido provocado por la transmisión de la imagen	17
2.4	Ruido provocado durante el procesamiento de la imagen	18
2.5	Supresión del ruido	19
2.5.1	Técnicas globales	19
2.5.1.1	Promediado en la adquisición de la imagen	19
2.5.1.2	Procesamiento en el dominio de la frecuencia	20
2.5.2	Técnicas locales o espaciales	22
2.5.2.1	Filtros lineales	22

2.5.2.2	Filtros no lineales	29
2.6	Estimación del ruido	33
3	Algoritmos de detección de bordes	37
3.1	Introducción	37
3.2	Tipos de bordes	39
3.2.1	Bordes en imágenes monocromáticas	39
3.2.2	Bordes en imágenes en color	41
3.3	Detección de bordes en imágenes monocromáticas	46
3.3.1	Algoritmos de detección de bordes monocromáticos	46
3.3.2	Fases en la detección de bordes en imágenes monocromáticas	48
3.3.3	Suavización de la imagen	50
3.3.4	Diferenciación de la imagen	51
3.3.4.1	Gradiente	53
3.3.4.2	Derivadas de segundo orden	54
3.3.4.3	Algoritmos basados en criterios de optimización .	59
3.3.5	Identificación de los bordes	67
3.3.5.1	Identificación de bordes en métodos basados en el gradiente	67
3.3.5.2	Identificación de bordes en métodos basados en las derivadas de segundo orden	73
3.3.6	Multidetectores y multiescalas	74
3.4	Detección de bordes en imágenes en color	76
3.4.1	Características de los algoritmos de detección de bordes en imágenes en color	76
3.4.1.1	Espacios de color utilizados por los algoritmos de detección de bordes	77
3.4.1.2	Suavización de imágenes en color	78

3.4.1.3	Fundamentos teóricos de los detectores de bordes en color	78
3.4.2	Detectores de bordes en color basados en técnicas monocromáticas	79
3.4.2.1	Detectores que no utilizan la información cromática de la imagen en color	79
3.4.2.2	Detectores basados en la reducción de la dimensionalidad de la imagen en color	80
3.4.2.3	Detectores basados en la fusión de bordes	81
3.4.2.4	Detectores basados en los gradientes monocromáticos	83
3.4.3	Detectores que consideran a las imágenes en color como campos vectoriales	86
3.4.3.1	Detectores de bordes en color basados en el ordenamiento de vectores	87
3.4.3.2	Detectores de bordes en color basados en el vector gradiente	93
3.4.3.3	Detectores de bordes en color basados en las derivadas de segundo orden	103
3.4.3.4	Detectores de bordes en color que utilizan la entropía	107
3.4.3.5	Detectores de bordes en color basados en técnicas de <i>clustering</i>	107
3.4.3.6	Detector de bordes en color basado en histogramas	108
3.4.3.7	Detector de bordes en color basado en la teoría bayesiana	109
3.4.3.8	Detector de bordes en color basado en polinomios ortogonales	109
4	Estudios comparativos de algoritmos de detección de bordes	111
4.1	Introducción	111
4.2	Características generales de los estudios comparativos	112

4.2.1	Propósito de la evaluación	112
4.2.2	Tipo de evaluación	113
4.2.3	Tipos de imágenes utilizados y tipos de bordes que contienen	116
4.2.4	Tipos de detectores evaluados	119
4.3	Evaluación de algoritmos de detección de bordes en imágenes monocromáticas	119
4.3.1	Calificaciones relativa y absoluta de los detectores de bordes	120
4.3.2	Evaluación realizada por Abdou y Pratt	122
4.3.3	Uso de las componentes necesarias de un espacio vectorial de errores para la evaluación de los detectores de bordes .	127
4.4	Evaluación de algoritmos de detección de bordes en imágenes en color	133
4.4.1	Estudio de las versiones de color del operador de Canny . .	134
4.4.2	Estudio realizado por Allen y Huntsberger	136
4.4.3	Comparación de tres algoritmos de segmentación en cuatro espacios de color	139
4.4.4	Comparación realizada por Trahanias y Venetsanopoulos .	141
4.4.5	Comparación realizada por Koschan	143
4.4.6	Estudio comparativo realizado por Zhu, Plataniotis y Venetsanopoulos	145
5	Algoritmos de adelgazamiento de bordes	155
5.1	Introducción	155
5.2	Transformación del eje “medial”	158
5.3	Algoritmo de Rosenfeld y Kak	158
5.4	Algoritmo de Zhang y Suen	161
5.5	Algoritmo de Wang y Zhang	162
5.6	Algoritmo de la imagen densa	164
5.7	Algoritmo de adelgazamiento “ $k \times k$ ”	168

5.7.1	Versión secuencial	173
5.7.2	Versión paralela	176
5.7.3	Variante para ventanas de 3×3 puntos	179

6 Reconocimiento de contornos 183

6.1	Introducción	183
6.2	Representación de un contorno	184
6.2.1	Representaciones basadas en los sistemas de coordenadas .	185
6.2.2	Código de cadena	187
6.2.3	Signatura	188
6.2.4	Método de la altura de arco	190
6.2.5	Aproximaciones poligonales	191
6.2.6	<i>Splines</i> y <i>B-splines</i>	193
6.2.7	Representaciones jerárquicas	194
6.3	Descriptores de un contorno	195
6.3.1	Atributos métricos	196
6.3.2	Atributos topológicos	198
6.3.3	Atributos analíticos	198
6.4	Algoritmos de reconocimiento de contornos	204

III Evaluación y Diseño 207

7 Análisis de las medidas de evaluación 209

7.1	Introducción	209
7.2	Porcentaje de discrepancia	210
7.3	Medida de Baddeley	211
7.4	Variantes o extensiones de las medidas de evaluación	212

7.4.1	Variante de la medida de Pratt	213
7.4.2	Versión general del método de Venkatesh y Kitchen	214
7.4.3	Normalización de las medidas de evaluación	217
7.5	Evaluación de las medidas de calidad	218
7.5.1	Diseño de las imágenes artificiales	218
7.5.2	Algoritmos de detección de bordes utilizados	220
7.5.3	Análisis de los resultados de las medidas de evaluación . .	222
8	Análisis de los detectores de bordes en color	229
8.1	Introducción	229
8.2	Algoritmos de detección de bordes en color evaluados	229
8.3	Pruebas realizadas con imágenes artificiales	231
8.4	Pruebas realizadas con imágenes reales	235
8.5	Análisis de los resultados	241
9	Diseño de un algoritmo de adelgazamiento de bordes	243
9.1	Introducción	243
9.2	Variante basada en la conectividad de orden 8	244
9.3	Variante basada en la conectividad de orden 4	247
9.4	Aplicación de los algoritmos	249
10	Extracción del contorno exterior	251
10.1	Introducción	251
10.2	Identificación de los puntos del contorno exterior	252
10.2.1	Método del sondeo	253
10.2.2	Método de la pila	254
10.2.3	Comparación entre los métodos	256
10.3	Extracción de los contornos de una imagen	257

10.4 Unión de los contornos de una imagen	264
10.5 Resumen de la metodología de extracción del contorno exterior . .	268
11 Algoritmo de comparación de contornos	271
11.1 Introducción	271
11.2 Parametrización de los contornos	272
11.3 Puntos críticos o relevantes de un contorno	275
11.3.1 Estimación de la curvatura en un punto: algoritmo de Mokhtarian y Mackworth	275
11.3.2 Puntos críticos utilizados en el algoritmo de comparación .	276
11.4 Algoritmo de Gu y Tjahjadi	278
11.5 Mejora del algoritmo de comparación	282
11.5.1 Función de comparación basada en el ajuste de los ángulos de los puntos críticos	284
11.5.2 Función de comparación basada en el ajuste de los ángulos y los radios de los puntos críticos	285
12 Análisis de la metodología de reconocimiento	289
12.1 Introducción	289
12.2 Fases de la metodología de reconocimiento	289
12.3 Pruebas de la metodología de reconocimiento	291
12.3.1 Análisis de las funciones de comparación	292
12.3.2 Estudio de la influencia de la orientación y el tamaño ini- ciales de los objetos	300
12.3.3 Pruebas con imágenes reales	304
12.4 Análisis de los resultados	313
IV Conclusiones y futuros estudios	315
13 Conclusiones y futuros estudios	317

13.1 Conclusiones	317
13.2 Futuros estudios	318
A Color	321
A.1 Definición y características	321
A.2 Espacios de color	324
A.2.1 Espacio de color <i>RGB</i>	326
A.2.2 Espacios de color <i>rgb</i> , <i>Irg</i> e <i>YT₁T₂</i>	327
A.2.3 Espacios de color <i>XYZ</i> y <i>xyz</i>	328
A.2.4 Espacios de color <i>L*u*v*</i> y <i>L*a*b*</i>	329
A.2.5 Espacios de color <i>YIQ</i> e <i>YUV</i>	331
A.2.6 Espacio de color <i>CMY</i>	332
A.2.7 Espacios de color <i>HSI</i> , <i>HLS</i> , <i>HSV</i> y <i>TekHVC</i>	333
A.2.8 Espacios de colores contrarios	336
A.3 Imagen digital en el espacio de color <i>RGB</i>	337
Bibliografía	341
Índice de Figuras	373
Índice de Tablas	385

Parte I

Introducción

Capítulo 1

Introducción

1.1 Procesamiento digital de imágenes

La visión permite a las personas conocer las posiciones e identidades de los objetos y las relaciones entre ellos, facilitando la realización de muy diferentes tareas inteligentes: planificación de procesos, investigación científica, control de sistemas, clasificación, etc.

La revolución tecnológica ha permitido diseñar cámaras y ordenadores cada vez más potentes y con unos precios más asequibles, lo cual ha provocado la aparición y desarrollo de “los sistemas de visión artificial”, que pretenden desarrollar dos actividades estrechamente relacionadas:

- “El análisis de imágenes digitales”, que tiene como objetivo la descripción y reconocimiento del contenido de una imagen digital.
- “La visión computacional”, que pretende dotar a los ordenadores de la capacidad de poder simular la visión humana.

Para poder realizar estas actividades, los sistemas de visión artificial necesitan utilizar el “procesamiento digital de imágenes”, el cual consiste, básicamente, en la transformación de una imagen del mundo real a un formato digital inteligible por un ordenador y su posterior procesamiento por parte de éste.

Los objetivos fundamentales del procesamiento digital de imágenes son:

- Mejorar la calidad visual de las imágenes para permitir la interpretación humana.
- Extraer información de las imágenes en un formato entendible por el ordenador.

El procesamiento digital de imágenes se ha convertido en una actividad multidisciplinar, ya que impulsa la colaboración de las Ciencias de la Computación con muchas áreas de investigación científica.

El procesamiento digital de imágenes consta de varias etapas (figura 1.1) que se ejecutan consecutivamente para obtener los resultados deseados a partir de la imagen original.

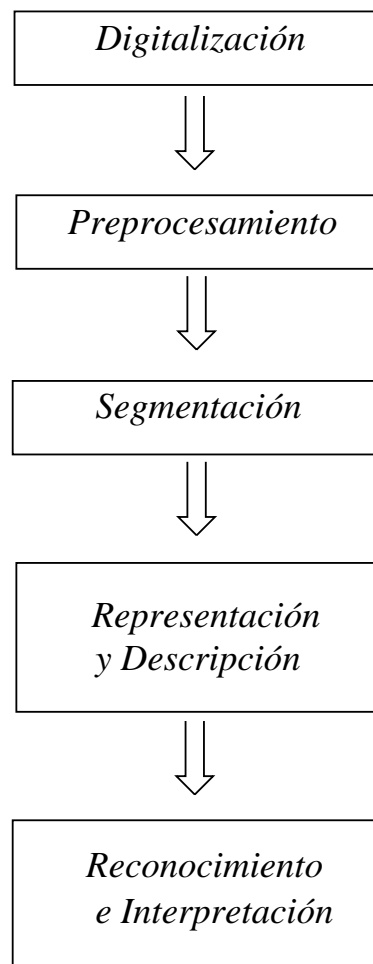


Figura 1.1: Fases del procesamiento digital de imágenes.

La primera etapa se denomina “digitalización” y consiste en captar una determinada imagen real mediante cualquier dispositivo adecuado (cámara de vídeo, *scanner*, ...) y, utilizando un conversor analógico-digital, realizar la conversión de la imagen captada a un formato digital inteligible por un ordenador.

En todo sistema de captura de imágenes se produce una cierta degradación de la imagen digitalizada respecto a la original, manifestándose esta degradación en forma de ruido y pérdida de definición de la imagen. Entre las posibles causas de esta degradación se encuentran los siguientes factores: calibración o enfoque de

la cámara defectuosos, ruido producido por los sensores de captura, movimiento del dispositivo de captura, transmisión defectuosa de la señal captada y perturbaciones aleatorias como la propagación de la radiación en el medio de transmisión (generalmente el aire).

A la digitalización le sigue una segunda etapa denominada “preprocesamiento” de la imagen, cuyo objetivo es contrarrestar la degradación introducida por el sistema de captura, de forma que las siguientes etapas del procesamiento digital de imágenes tengan una probabilidad de éxito mayor que la que tendrían si manipularan directamente la imagen degradada. Las operaciones típicas de esta etapa son la supresión de ruido y el realce del contraste.

La “segmentación” constituye la tercera etapa y tiene como objetivo fundamental la extracción de la información contenida en la imagen. Dicha extracción se realiza mediante una descomposición de la imagen en unidades o partes que son homogéneas con respecto a una o más características y que tienen una fuerte relación con objetos o áreas del mundo real.

Cuando se alude a las partes u objetos componentes de una imagen, hay que aclarar que dichos objetos dependen de la aplicación que se le vaya a dar al proceso de segmentación. Así, por ejemplo, en un sistema de reconocimiento aéreo de objetivos militares, los objetos de interés serían vehículos, edificaciones, rutas de transporte, etc. Sin embargo, para un estudio geográfico, la misma imagen tendría otros objetos de interés, como podrían ser los accidentes montañosos y las vías fluviales. Por consiguiente, cada objeto de la imagen segmentada debe ser etiquetado de forma unívoca para que pueda ser integrado dentro de una descripción estructurada de la imagen original.

La salida generada por la segmentación suele ser una imagen binaria. Para que el ordenador pueda seguir realizando las siguientes etapas del procesamiento digital de imágenes, se hace necesario describirle la salida del proceso de segmentación de una forma adecuada. Por consiguiente, la imagen binaria resultante del proceso de segmentación representa o bien el contorno de los objetos hallados, o bien la región que abarca cada objeto. Esta etapa se denomina “representación” y viene acompañada de otra, la “descripción”, cuya finalidad es extraer información de la representación elegida, obteniéndose las características o los descriptores que permitirán la posterior clasificación de los objetos. Entre estos descriptores se encuentran el perímetro del contorno, el área, el número de huecos, los puntos más significativos o relevantes, etc.

Por último, existen dos etapas denominadas “reconocimiento” e “interpretación”. El reconocimiento se encarga de etiquetar o clasificar los diferentes objetos de la imagen utilizando sus descriptores, de forma que aquellos objetos detectados que presenten unos descriptores semejantes se agrupan en una misma clase. Finalmente, la interpretación se encarga de darle un significado a los grupos de objetos reconocidos.

1.2 Reconocimiento de objetos

El reconocimiento de objetos trata de identificar o clasificar un objeto mediante el análisis de sus características. El reconocimiento de objetos forma parte de muchas aplicaciones de inteligencia artificial, como son los sistemas de navegación de robots móviles, detección de tumores, análisis celular, reconocimiento de caracteres, identificación de objetivos militares, investigación criminal, etc.

El procesamiento digital de imágenes permite el reconocimiento de objetos tridimensionales (*objetos 3D*) mediante el análisis de sus proyecciones bidimensionales (*objetos 2D*). El reconocimiento de objetos bidimensionales está justificado por los siguientes motivos (Mokhtarian 1995):

- Un objeto tridimensional (*3D*), cuando es situado en una superficie plana y es visualizado por una cámara fija, tiene un número limitado de posiciones estables, cada una de las cuales pueden ser modelada usando un contorno bidimensional (*2D*).
- Incluso cuando una imagen contiene un único objeto, el problema de su reconocimiento puede llegar a ser desafiante debido a las formas arbitrarias de los objetos, el ruido o a deformaciones locales de los objetos provocadas por la perspectiva, errores de segmentación o la naturaleza de los objetos.

Muchos algoritmos de reconocimiento de objetos están basados exclusivamente en el análisis de la forma de los mismos. Se puede definir la “forma” como la silueta generada por un objeto cuando es iluminado por una fuente de luz situada a una distancia infinita (Loncaric 1998).

Los métodos de análisis de formas se pueden clasificar en dos grandes grupos atendiendo al tipo de puntos del objeto que utilicen:

- Métodos basados en el contorno o métodos externos: sólo utilizan los puntos que forman parte del contorno (frontera o bordes) del objeto.
- Métodos globales o métodos internos: utilizan todos los puntos del objeto, tanto los pertenecientes al contorno como los interiores.

El contorno de un objeto puede obtenerse mediante la aplicación de un algoritmo de segmentación. La segmentación es una de las tareas más difíciles y complicadas del procesamiento digital de imágenes, debido, fundamentalmente, a las siguientes causas (Rosenfeld y Kak 1982a; Sonka et al. 1993):

- La ambigüedad de la imagen, ya que, en la práctica, los objetos no están siempre claramente definidos.

- La presencia de ruido, que puede dificultar la correcta percepción de los objetos de la imagen.

El ruido es un factor que ejerce una gran influencia en los sistemas de visión digital y es necesario conocer su origen, características y los métodos que permiten eliminarlo o atenuarlo.

El intentar desarrollar un algoritmo de segmentación que pueda ser válido para todas las clases de imágenes y que sea útil para todas las aplicaciones científicas es un reto muy interesante. Este reto tiene su primera y principal dificultad en la selección de la técnica apropiada para diseñar el algoritmo.

La segmentación de imágenes puede ser llevada a cabo desde dos perspectivas filosóficas diferentes (Castleman 1979):

- Orientada a las regiones: utiliza las propiedades de homogeneidad de las regiones para distinguir a los objetos entre sí.
- Orientada a los bordes: intenta localizar los límites, contornos o fronteras que existen entre las regiones para obtener los objetos de la imagen.

Las técnicas de segmentación de detección de bordes y orientadas a regiones constituyen estrategias válidas y complementarias. De hecho, se ha de indicar que la integración de ambas técnicas ha sido realizada en múltiples ocasiones (Bajcsy et al. 1986; Chakarborty y Duncan 1999; Gambotto 1993; Lemoigne y Tilton 1995; Moghaddamzadeh y Bourbakis 1997; Pavlidis y Liow 1990; Perkins 1980).

La detección de bordes ha sido la técnica de segmentación más utilizada por los algoritmos de reconocimiento de objetos basados en el análisis de los contornos (van Otterloo 1991). Los motivos que justifican tal elección son:

- El principal objetivo de la detección de bordes es localizar la frontera o contorno de los objetos.
- El coste computacional de los algoritmos de detección de bordes es menor que el requerido por los algoritmos de segmentación orientados a regiones.

Se han propuesto numerosos detectores de bordes en imágenes monocromáticas (Davis 1975; Faugeras 1993; Fu y Mui 1981; González y Woods 1996; Levine 1985; Matsuyama 1989; Nalwa 1993; Pal y Pal 1993; Pitas 1993; Pratt 1978; Rosenfeld y Kak 1982a; Sonka et al. 1993; Torre y Poggio 1986; Zamperoni 1995; Ziou y Tabbone 1998); por el contrario, el número de algoritmos de detección de bordes en imágenes en color es comparativamente mucho menor (Skarbek y Koschan 1994).

La utilización del color en el procesamiento digital de imágenes está justificada por diferentes razones, entre las que cabe destacar las siguientes:

- En el análisis automático de imágenes, el color representa un potente descriptor que facilita la identificación de un objeto y su extracción de una escena (González y Woods 1996).
- Una imagen en color suministra más información que una imagen monocromática. Se ha comprobado que los bordes detectados en una imagen monocromática representan el 90% de los bordes detectados en una imagen en color (Novak y Shafer 1987). Hay, por tanto, un 10% de bordes no detectados en una imagen monocromática que pueden ser importantes para una tarea de visión artificial, como puede ser el reconocimiento de objetos.
- No se puede detectar ningún borde en imágenes con niveles de gris cuando objetos vecinos tienen diferentes tonos (*hue*) pero iguales intensidades (Koschan 1995).
- La detección de bordes en imágenes en color también supera a la detección de bordes en imágenes monocromáticas cuando las imágenes tienen un contraste bajo (Koschan 1995; Zhu et al. 1999).
- El progreso tecnológico (cámaras, procesadores, memorias) permite investigar más fácil y eficientemente en el procesamiento de imágenes en color (Chapron 1992).

La utilización de un detector de bordes en imágenes en color es una opción muy interesante y, por tanto, se ha de determinar cuál es el algoritmo más eficiente.

Sin embargo, antes de comparar los detectores de bordes en color, es necesario elegir la medida de calidad más adecuada para realizar tal evaluación. Se han propuesto y realizado numerosos métodos de evaluación de detectores de bordes en imágenes monocromáticas, mientras que sólo se ha tenido conocimiento de cuatro estudios de detectores de bordes en color (Koschan 1995; Novak y Shafer 1987; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999).

Se considera, por tanto, necesario realizar un estudio comparativo que permita seleccionar la medida de calidad más adecuada para, posteriormente, evaluar los algoritmos de detección de bordes en imágenes en color.

Una vez seleccionado el algoritmo de detección de bordes en imágenes en color, se procede a su aplicación para obtener los contornos de los objetos de la imagen analizada.

El contorno de un objeto ha de tener el grosor de un punto para facilitar las fases posteriores de descripción y reconocimiento. Para obtener el contorno con un punto de grosor es necesario aplicar un algoritmo que adelgace los bordes obtenidos por el detector. Debido a ello, se han de describir las características de los algoritmos de adelgazamiento de bordes existentes.

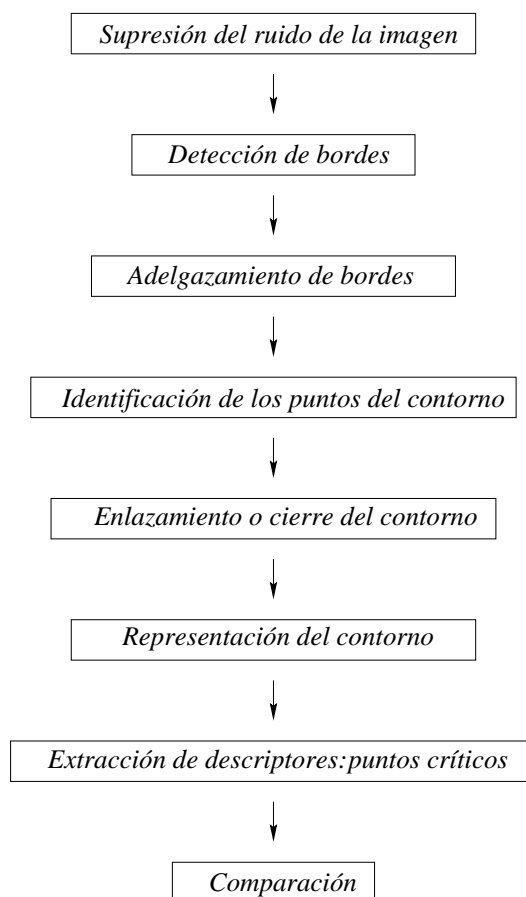


Figura 1.2: Fases de un algoritmo de reconocimiento de objetos basado en el uso de los puntos críticos de los contornos.

La identificación de los puntos que forman parte de un contorno también tiene que ser tenida en cuenta, ya que la imagen generada por el detector puede contener puntos espurios provocados por el ruido, sombras, destellos, etc. La correcta identificación de los puntos del contorno es necesaria para evitar problemas de reconocimiento.

Podría ocurrir que el contorno obtenido no fuese cerrado o que estuviese dividido en varios trozos, lo cual supone un problema adicional de cierta complejidad que ha de ser abordado, puesto que muchos algoritmos de reconocimiento de objetos requieren que el contorno esté cerrado completamente. La aplicación de un algoritmo de enlazamiento permitirá la conexión entre los trozos del contorno y su cierre.

El reconocimiento de objetos mediante el análisis de los contornos necesita un sistema de representación que sea invariante a rotaciones o traslaciones y que facilite la extracción de los descriptores o características esenciales de los contornos.

Los algoritmos de reconocimiento más eficientes no utilizan todos los puntos del contorno, puesto que su coste computacional sería inaceptable. Sólo necesitan aquellos puntos que sean más significativos, críticos o relevantes, los cuales son utilizados como descriptores del contorno.

Se han propuesto diferentes algoritmos para extraer los puntos críticos o relevantes de un contorno, algunos de los cuales realizan un análisis de la curvatura de los puntos utilizando el espacio de escalas gaussiano.

Finalmente, los descriptores del contornos son utilizados por un algoritmo de comparación que determina si el objeto analizado pertenece o no a una determinada clase o categoría.

El diseño de una metodología de reconocimiento de objetos basada en el análisis de los puntos críticos de los contornos consta de todas las fases que se han enumerado (figura 1.2).

1.3 Objetivos

El presente trabajo trata de estudiar las características esenciales de las fases de una metodología de reconocimiento de objetos basada en la detección de bordes en imágenes en color y en el análisis de la curvatura de los puntos de los contornos de los objetos. Dicho estudio consistirá en una revisión del “estado del arte” de cada una de las fases y en el diseño de algoritmos que mejoren su rendimiento.

Este objetivo principal se puede descomponer en los siguientes objetivos específicos:

- Estudio de la influencia, el origen y la supresión del ruido en la imágenes digitales.
- Descripción de los tipos de bordes y de los algoritmos de detección de bordes, tanto en imágenes monocromáticas como en color.
- Análisis de los métodos de evaluación de los algoritmos de detección de bordes.
- Comparación de diferentes medidas de evaluación de detectores de bordes para elegir la que tenga un mejor rendimiento.
- Uso de la medida de evaluación seleccionada para comparar los algoritmos de detección de bordes en color y elegir el más eficiente.
- Estudio de las características de los algoritmos de adelgazamiento de bordes.

- Diseño de un método que permita extraer el contorno exterior de un objeto contenido en una imagen binaria.
- Revisión de los sistemas de representación y de los métodos de descripción de contornos utilizados por los algoritmos de reconocimiento.
- Obtención de los descriptores de un contorno: análisis de la curvatura de los puntos para obtener los puntos críticos o más significativos.
- Diseño de una función de comparación de contornos que utilice los puntos críticos.

1.4 Organización del documento

El resto del presente documento ha sido organizado en tres partes:

- Revisión de los antecedentes.
- Evaluación y diseño.
- Conclusiones y futuros estudios.

La revisión de antecedentes consta de:

- Un estudio sobre la influencia, el origen y los métodos de supresión o atenuación del ruido en imágenes digitales (capítulo 2).
- Una descripción de los tipos de bordes y de los algoritmos de detección de bordes, tanto en imágenes monocromáticas como en color (capítulo 3).
- Un análisis de los estudios comparativos de los algoritmos de detección de bordes (capítulo 4).
- Una descripción del adelgazamiento de bordes y de algunos de los algoritmos más interesantes (capítulo 5).
- Una revisión de los métodos de representación y descripción de contornos más importantes (capítulo 6).

La parte destinada a la evaluación y diseño describe todas las pruebas realizadas y los algoritmos diseñados para el desarrollo de una metodología de reconocimiento de objetos basada en la detección de bordes en imágenes en color y en el análisis de la curvatura de los puntos de los contornos. Esta parte se ha dividido en los siguientes capítulos:

- El capítulo 7 describe cómo se han generalizado algunas medidas para poder utilizar bordes con cualquier forma geométrica. Dichas medidas han sido comparadas entre sí para seleccionar la medida más adecuada para la evaluación de algoritmos de detección de bordes en imágenes en color.
- El capítulo 8 describe las pruebas realizadas y los resultados obtenidos al comparar algunos algoritmos de detección de bordes en imágenes en color.
- En el capítulo 9 se proponen dos variantes de un algoritmo de adelgazamiento de bordes.
- El capítulo 10 describe el método que se ha diseñado para extraer el contorno exterior de un objeto.
- La obtención de los puntos críticos de un contorno mediante el análisis de los puntos de curvatura extrema es descrita en el capítulo 11. También se incluye un algoritmo de comparación basado en los puntos críticos y dos variantes que mejoran su rendimiento.
- El capítulo 12 describe las pruebas realizadas para analizar el rendimiento de la metodología de reconocimiento de objetos propuesta.

La última parte consta del capítulo 13 que resume las conclusiones del trabajo realizado y propone algunas futuras líneas de investigación.

Además, el documento incluye un apéndice en el que se describen las características de los espacios de color más utilizados en el procesamiento digital de imágenes (apéndice A).

Parte II

Revisión de antecedentes

Capítulo 2

Ruido

2.1 Introducción

El procesamiento digital de una imagen ha de tener en cuenta la presencia de algunos errores que pueden degradarla. Esta degradación se suele denominar “ruido”.

El ruido es un factor que influye notablemente en el diseño de los sistemas de análisis de imágenes, ya que, en la medida de lo posible, tales sistemas deben ser robustos, es decir, han de generar los mismos resultados que se obtendrían cuando el ruido estuviera ausente (Billingsley 1979).

El ruido suele ser descrito por sus características probabilísticas, de forma que puede agruparse en dos clases principales:

- “Ruido local determinístico”: se caracteriza por presentar una forma y apariencia constante para cada sistema en particular. A menudo se manifiesta como una distribución constante y determinada dentro de la imagen. Un ejemplo típico es la aparición en la imagen de una franja con unos niveles de intensidad sensiblemente diferentes a los del resto de la imagen.
- “Ruido aleatorio”: es producido por fuentes imprevistas y, en contraposición al ruido determinístico, su distribución es aleatoria y cambiante con el tiempo.

El ruido se puede producir durante la adquisición, la transmisión o el procesamiento de la imagen.

En algunas ocasiones, la imagen puede contener un ruido intrínseco que está provocado por la textura de los objetos o por un entorno especial en el que se

sitúa la escena (v.g.: imágenes subacuáticas o con presencia de humo o vapor). Este tipo de error se denomina “ruido semántico”.

2.2 Ruido provocado por la adquisición de la imagen

Los dispositivos de adquisición de una imagen puede ser divididos en dos categorías (Pitas et al. 1990; Plataniotis y Venetsanopoulos 2000):

- Sensores fotoquímicos: películas o diapositivas fotográficas.
- Sensores fotoelectrónicos: tubos *vidicon*, dispositivos de inyección de carga (CID, *Charge Injection Devices*) y dispositivos de carga acoplada (CCD, *Charge Coupled Devices*).

Aunque los dispositivos fotoquímicos pueden detectar y grabar la imagen al mismo tiempo, la imagen que producen no puede ser fácilmente digitalizada. El ruido introducido por estos dispositivos puede ser de dos clases:

- Ruido de “grano de película” (*film grain noise*): este ruido es provocado por los granos de plata que precipitan durante la exposición de la película y que son usados para la emulsión fotosensitiva. Este tipo de ruido es muy dominante cuando las imágenes son adquiridas con una velocidad alta. Este ruido puede ser modelado mediante distribuciones de Poisson o gaussianas (Plataniotis y Venetsanopoulos 2000; Sonka et al. 1993).
- Ruido provocado por el polvo que pueden tener los componentes ópticos o por el polvo presente durante el revelado de las películas.

Los dispositivos fotoelectrónicos superan a las películas en que pueden digitalizar la imagen al mismo tiempo que la adquieren. Se pueden diferenciar dos tipos de ruido en las imágenes adquiridas mediante sensores fotoelectrónicos (Jähne 1993; Michelli et al. 1989; Plataniotis y Venetsanopoulos 2000; Pratt 1978):

- Ruido térmico (*thermal noise*): es provocado por el calor originado por los componentes electrónicos. Este tipo de ruido suele ser modelado como ruido aditivo blanco gaussiano de media cero.
- Ruido fotoelectrónico: es producido por la fluctuación aleatoria del número de fotones de la luz. Si la fluctuación es baja, este ruido puede ser modelado

mediante una distribución de Poisson; en caso contrario, cuando la fluctuación es alta, este ruido puede ser modelado como un proceso gaussiano con desviación típica igual a la raíz cuadrada de la media.

Existe otro tipo de ruido estrechamente relacionado con la adquisición: “el ruido de cuantificación”. Este ruido ocurre cuando son usados insuficientes niveles de cuantificación para digitalizar la imagen. En tales situaciones aparecen contornos falsos. Este tipo de ruido se puede eliminar mediante una transformación en la escala de niveles de intensidad (por ejemplo, ecualización del histograma de frecuencias).

2.3 Ruido provocado por la transmisión de la imagen

La transmisión de una imagen digital entre dos dispositivos remotos (una cámara y un ordenador, dos ordenadores o un monitor y un ordenador) puede incrementar el ruido de la imagen. Cuando una imagen es transmitida a través de algún canal, se produce un ruido que generalmente es independiente de la señal de la imagen. La degradación independiente de la señal se denomina “ruido aditivo” y puede ser descrito por la siguiente ecuación

$$g(x, y) = f(x, y) + \eta(x, y) \quad (2.1)$$

donde la imagen original f y el ruido η son dos variables independientes.

Por el contrario, existen otras situaciones en las que la magnitud del ruido sí depende de la magnitud de la señal. Si la magnitud del ruido es mucho mayor que la de la señal, entonces la ecuación 2.1 se transforma en

$$g(x, y) = f(x, y) + \eta(x, y)f(x, y) = f(x, y)(1 + \eta(x, y)) \approx f(x, y)\eta(x, y) \quad (2.2)$$

La ecuación 2.2 describe “el ruido multiplicativo”. Un ejemplo de ruido multiplicativo es el que se produce en la pantalla del televisor. Este ruido depende de la líneas del televisor: en el área de una línea, este ruido es máximo, y entre dos líneas es mínimo (Sonka et al. 1993).

Existe un tipo de error que también aparece con frecuencia durante la transmisión de la imagen: “el ruido impulsivo” o “ruido de sal y pimienta” (*salt-pepper noise*). Este ruido se caracteriza por la presencia de puntos de la imagen que poseen unos niveles de intensidad que difieren bastante de los niveles de los puntos de su entorno. Frecuentemente, el ruido de sal y pimienta aparece sobre la imagen en forma de impulsos blancos o negros. La ecuación 2.3 muestra este tipo de ruido:

$$g(x, y) = \begin{cases} \eta(x, y) & \text{con probabilidad } p \\ f(x, y) & \text{con probabilidad } 1 - p \end{cases} \quad (2.3)$$

Los impulsos de ruido son denotados por $\eta(x, y)$ y aparecen con probabilidad p . Los impulsos pueden tener valores prefijados (v.g.: 0 ó 255) o pueden tener una distribución de probabilidad de “cola larga” (*long-tailed*). La distribución laplaciana es usada frecuentemente como modelo de distribución de probabilidad de cola larga (Pitas 1993).

2.4 Ruido provocado durante el procesamiento de la imagen

En muchos casos, las imágenes digitales pueden ser corrompidas artificialmente durante el procesamiento para comprobar el rendimiento o comportamiento de los algoritmos en presencia de ruido.

Además del ruido impulsivo descrito en la ecuación 2.3, también se genera o introduce ruido aditivo (2.1) o multiplicativo (2.2).

Uno de los tipos de ruido que más se suele utilizar es el ruido “blanco” (*white noise*), que se caracteriza por tener una potencia espectral constante, es decir, su intensidad no decrece cuando aumenta la frecuencia. Este ruido tiene de media cero y no está correlado. El ruido blanco es una aproximación ideal del ruido que puede degradar las imágenes, siendo su ventaja principal que su uso simplifica los cálculos (González-Jiménez 2000; Sonka et al. 1993).



Figura 2.1: (a) Imagen original; (b) imagen con ruido aditivo gaussiano ($\sigma = 10$).

Otro tipo de ruido frecuentemente usado es “el ruido gaussiano”, cuya función de densidad de probabilidad viene expresada por la curva gaussiana. La versión unidimensional de la curva gaussiana es definida como

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.4)$$

donde μ es la media y σ es la desviación típica de la variable aleatoria. El ruido gaussiano es una muy buena aproximación al ruido que ocurre en muchos casos

prácticos. El ruido gaussiano de media cero se suele utilizar para generar ruido aditivo blanco.

Otro tipo de ruido aleatorio que se utiliza en el procesamiento digital de imágenes es “el ruido exponencial”, cuya función de densidad de probabilidad es

$$p(x) = \lambda e^{-\lambda x} \quad \text{para } x \geq 0 \quad (2.5)$$

donde la media y la varianza de la distribución son, respectivamente, $\frac{1}{\lambda}$ y $\frac{1}{\lambda^2}$.

2.5 Supresión del ruido

Las técnicas de supresión del ruido están estrechamente relacionadas con los algoritmos de suavizado y mejora de la imagen:

- Técnicas globales: operan sobre toda la imagen en conjunto.
 - Promediado en la adquisición de la imagen.
 - Procesamiento en el dominio de la frecuencia.
- Técnicas locales o espaciales: analizan de forma individualizada el comportamiento de cada punto de la imagen.
 - Filtros lineales.
 - Filtros no lineales.

Aunque todas las técnicas suprimen el ruido satisfactoriamente, se prefieren los filtros espaciales, ya que tienen, en general, un mejor rendimiento con un menor coste en memoria y en tiempo de ejecución. Sin embargo, no existe un único filtro lineal o no lineal que sea óptimo para todas las imágenes (Maravall 1993; Michelli et al. 1989).

2.5.1 Técnicas globales

2.5.1.1 Promediado en la adquisición de la imagen

Una imagen digital con ruido $g(x, y)$ puede ser caracterizada por la adición de una función de ruido $\eta(x, y)$ a una imagen original $f(x, y)$, es decir,

$$g(x, y) = f(x, y) + \eta(x, y) \quad (2.6)$$

donde se supone que en cada punto (x, y) el ruido es una función sin correlación y tiene un valor de media cero (ruido blanco). El objetivo es reducir los efectos

del ruido mediante la media aritmética de un conjunto de imágenes $\{g_i(x, y)\}$ (González y Woods 1996; González-Jiménez 2000):

$$\bar{g}(x, y) = \frac{1}{N} \sum_{i=1}^N g_i(x, y) \quad (2.7)$$

obteniéndose que

$$E[\bar{g}(x, y)] = f(x, y) \quad (2.8)$$

y

$$\sigma_{g(x, y)}^2 = \frac{1}{N} \sigma_{\eta(x, y)}^2 \quad (2.9)$$

donde $E[\bar{g}(x, y)]$ es valor esperado de \bar{g} y $\sigma_{g(x, y)}^2$ y $\sigma_{\eta(x, y)}^2$ son las varianzas de \bar{g} y η en el punto (x, y) , respectivamente. La desviación típica en cada punto de la imagen promediada es

$$\sigma_{\bar{g}(x, y)} = \frac{1}{\sqrt{N}} \sigma_{\eta(x, y)} \quad (2.10)$$

Las ecuaciones 2.9 y 2.10 indican que, conforme aumenta N , la variabilidad de los valores de la imagen en cada punto (x, y) decrece. Puesto que $E[\bar{g}(x, y)] = f(x, y)$, la función $\bar{g}(x, y)$ se aproxima a $f(x, y)$ conforme el número de imágenes con ruido empleadas en el proceso de promediado aumenta. En la práctica, las imágenes $g_i(x, y)$ deben grabarse con el fin de evitar la falta de nitidez de la imagen de salida (González y Woods 1996).

2.5.1.2 Procesamiento en el dominio de la frecuencia

En la mayoría de las situaciones, el ruido se caracteriza por contribuir significativamente a las altas frecuencias de la transformada discreta de Fourier de la imagen (Maravall 1993).

Si $f(x, y)$ es una función que representa a una imagen digital de $N \times N$ puntos, entonces su transformada discreta de Fourier se define como

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{\frac{-2\pi i}{N}(ux+vy)} \quad (2.11)$$

El procesamiento en el dominio de la frecuencia elimina el ruido mediante la aplicación del “filtro paso bajo” a la transformada de Fourier de la imagen:

$$G(u, v) = H(u, v)F(u, v) \quad (2.12)$$

donde

$$H(u, v) = \begin{cases} 1 & \text{si } D(u, v) \leq D_0 \\ 0 & \text{si } D(u, v) > D_0 \end{cases} \quad (2.13)$$

siendo D_0 una cantidad no negativa y $D(u, v)$ la distancia desde el punto (u, v) al origen del plano de frecuencias:

$$D(u, v) = \sqrt{u^2 + v^2} \quad (2.14)$$

Este filtro es radialmente simétrico respecto del origen de coordenadas y es suficiente mostrar una sección del mismo para poder especificarlo, tal y como se puede observar en la figura 2.2.

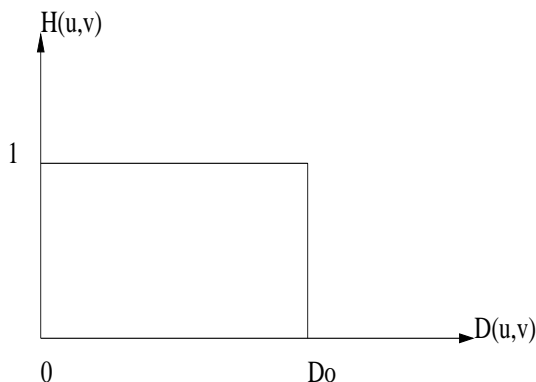


Figura 2.2: Filtro paso bajo en el dominio de la frecuencia.

El filtro paso bajo en el dominio de la frecuencia tiene como gran inconveniente el alto coste computacional que requiere.

Además del filtro paso bajo, la transformada discreta de Fourier es utilizada para definir otros filtros que permiten realzar la imagen en el dominio de la frecuencia. Uno de los más importantes es “el filtro de Wiener”, el cual está especialmente diseñado para mejorar imágenes borrosas o con ruido aditivo (Jain 1989; Pitas 1993).

El filtro de Wiener permite obtener unos resultados espectaculares en la mayoría de los casos (González y Woods 1996). Sin embargo, el alto coste computacional que requiere y la necesidad de conocer previamente las características del ruido hacen que la utilización de este filtro sea prácticamente inviable.

Los filtros que operan en el dominio de la frecuencia están relacionados con los filtros espaciales, puesto que la multiplicación de dos señales en el dominio de la frecuencia es equivalente a la convolución de éstas en el dominio espacial:

$$G(W) = F(W)H(W) \Leftrightarrow g(x) = f(x) \otimes h(x) \quad (2.15)$$

donde $G(W)$, $F(W)$ y $H(W)$ son, respectivamente, las transformadas de Fourier de las señales unidimensionales $g(x)$, $f(x)$ y $h(x)$. En el caso de imágenes digitales, estas funciones son bidimensionales y discretas (González-Jiménez 2000).

El procesamiento de imágenes en el dominio de la frecuencia no suele ser utilizado debido al alto coste computacional que requiere la transformada rápida de Fourier (*FFT*) en comparación con los filtros espaciales.

2.5.2 Técnicas locales o espaciales

Las técnicas locales también se denominan técnicas espaciales porque asignan a cada punto de la imagen un valor que depende de los valores de sus puntos vecinos.

Los filtros espaciales pueden ser clasificados como:

- Filtros lineales: el valor asignado a un punto de la imagen se obtiene mediante una combinación lineal de los valores de sus puntos vecinos.
- Filtros no lineales: utilizan un método no lineal para combinar los valores de los puntos vecinos.

La relativa sencillez de los filtros lineales ha favorecido su utilización en el procesamiento digital de imágenes. Desafortunadamente, la técnicas lineales no pueden hacer frente a las características no lineales del modelo de formación de la imagen digital y tampoco tienen en cuenta la naturaleza no lineal del sistema visual humano (Pitas et al. 1990; Plataniotis y Venetsanopoulos 2000).

Se ha probado que los filtros no lineales son más potentes que los lineales porque eliminan cierta clase del “mejor” ruido (ruido impulsivo), preservando, al mismo tiempo, la información del borde (Pitas y Venetsanopoulos 1986). Sin embargo, los filtros lineales suelen ser los más usados, especialmente por los algoritmos de detección de bordes.

Los filtros espaciales usan generalmente ventanas de tamaño y forma fija: ventanas cuadradas de 3×3 , 5×5 ó 7×7 puntos. Sin embargo, se pueden utilizar ventanas de tamaño y forma variable, no sólo para la imagen en su conjunto, sino para cada punto individual de la misma. Esta técnica persigue obtener el vecindario más homogéneo para cada punto. Debido a exigencias de cómputo y a la no existencia de una aproximación unificadora, la utilización de filtros espaciales con vecindarios adaptados a cada punto es poco frecuente (Sonka et al. 1993).

2.5.2.1 Filtros lineales

Los filtros espaciales lineales también se denominan “filtros del promediado del entorno” porque el valor que asignan a cada punto es obtenido mediante una

media ponderada de los valores de sus puntos vecinos. Se pueden definir diferentes filtros dependiendo de la forma de elección de los puntos vecinos y de la ponderación que se le da a cada uno de ellos.

Los filtros basados en el promediado del entorno permiten atenuar la influencia del ruido, pero tienen como principal inconveniente el enturbiamiento de la imagen, provocando que los bordes se difuminen, sobre todo cuanto aumenta el tamaño de la máscara de convolución (7×7 o mayores). Debido a ello, este tipo de filtros deben ser usados con precaución.

Los filtros basados en el promediado del entorno pueden ser clasificados como (Faugeras 1993; González-Jiménez 2000; Torre y Poggio 1986):

- Filtros de banda limitada: pueden eliminar ruido del cual se conoce su frecuencia. Estos filtros tienen una respuesta de impulso infinita (*infinite impulse response, IIR*) y, por tanto, actúan sobre una extensión infinita de la imagen presentando serios inconvenientes de implementación. Un ejemplo de este tipo de filtros sería el filtro “paso bajo ideal”.
- Filtros de soporte limitado o filtros de respuesta de impulso finita (*finite impulse response, FIR*) que actúan sobre una región finita de la imagen. Aunque interesantes desde un punto de vista computacional, presentan el inconveniente de actuar sobre una extensión infinita de frecuencias. Un ejemplo de este tipo de filtros es el filtro de “la media simple”.
- Filtros que minimizan la incertidumbre: se pueden considerar como una combinación entre las dos primeras clases de filtros, ya que optan por un compromiso entre la actuación en el dominio de la frecuencia y el espacial.

Los filtros lineales asumen que todos los puntos de la imagen aportan una información útil, lo cual no es cierto cuando el valor de un punto está completamente distorsionado por un error. Los filtros lineales no eliminan esta información errónea, sino que “la reparten” entre los demás puntos. La única solución consiste en detectar tales puntos ruidosos y eliminarlos.

Se van a describir dos tipos de filtros basados en el promediado del entorno:

- Filtros de “paso bajo”.
- Filtros que minimizan la incertidumbre.

Filtros de “paso bajo”

Los puntos ruidosos de una imagen suelen aparecer con unos niveles de intensidad muy diferentes a los de sus vecinos. Este hecho permite aplicar una técnica local

de supresión de ruido muy simple, que en el caso de una imagen monocromática viene expresada por la siguiente ecuación

$$g(x_0, y_0) = \begin{cases} \frac{1}{8} \sum_{i=1}^8 f(x_i, y_i) & \text{Si } |f(x_0, y_0) - \frac{1}{8} \sum_{i=1}^8 f(x_i, y_i)| > \epsilon \\ f(x_0, y_0) & \text{En caso contrario} \end{cases} \quad (2.16)$$

(x_8, y_8)	(x_1, y_1)	(x_2, y_2)
(x_7, y_7)	(x_0, y_0)	(x_3, y_3)
(x_6, y_6)	(x_5, y_5)	(x_4, y_4)

Figura 2.3: Vecindario de un punto (x_0, y_0) .

Como se puede observar en la ecuación 2.16, si la intensidad en un punto (x_0, y_0) difiere de la intensidad media de los puntos de su contorno en una cantidad superior a un umbral $\epsilon > 0$, entonces se le asigna dicho valor medio; en caso contrario, no se modifica el valor del punto (x_0, y_0) .

Debido a que la distribución del ruido dentro de la imagen suele ser aleatoria, el proceso anterior se puede simplificar mediante la aplicación directa de un filtro espacial de convolución, como el filtro de “paso bajo” (Pratt 1978).

Los filtros de “paso bajo” sustituyen el valor $f(x_0, y_0)$ en un punto (x_0, y_0) de la imagen por la media ponderada de su valor y de los valores de los puntos situados en su entorno:

$$g(x_0, y_0) = \sum_{i=0}^{N-1} w_i f(x_i, y_i) \quad (2.17)$$

donde los pesos han de cumplir las siguientes dos restricciones:

$$\sum_{i=0}^{N-1} w_i = 1 \quad (2.18)$$

y

$$w_i \geq 0, \quad \forall i \in \{0, 1, \dots, N-1\} \quad (2.19)$$

Se suele utilizar una máscara o ventana cuadrada de $N = n \times n$ puntos para definir el entorno o vecindario en el que se aplicará el filtro.

$$\begin{bmatrix} w_8 & w_1 & w_2 \\ w_7 & w_0 & w_3 \\ w_6 & w_5 & w_4 \end{bmatrix}$$

Figura 2.4: Máscara de convolución genérica de 3×3 puntos.

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(a) (b)

Figura 2.5: Máscaras de convolución del filtro espacial “paso bajo” de (a) 3×3 y (b) 5×5 puntos .

Dependiendo de los valores de N y de w_i , se pueden definir diferentes filtros espaciales de “paso bajo”. Los filtros más simples se obtienen cuando el tamaño de la máscara de convolución es $N = 3 \times 3$ ó $N = 5 \times 5$ puntos y el valor de cada peso $w_i = \frac{1}{N}$ (figura 2.5).

Una variante de este tipo de filtros son los filtros binomiales bidimensionales que pueden ser definidos a partir de la composición de filtros horizontales y verticales unidimensionales (Jähne 1993). El representante más simple de este tipo de filtros se define como:

$$B^2 = \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{4} [1 \quad 2 \quad 1] = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.20)$$

También se han propuesto otros filtros que sólo tienen en cuenta determinados puntos del entorno, evitando de esta manera que los bordes se difuminen. Algunos de estas variantes son:

- Filtros del vecino más próximo (Pietikäinen y Harwood 1986).
- Filtros basados en el uso de bloques de puntos vecinos (González-Jiménez 2000; Moghaddamzadeh y Bourbakis 1997).

Uno de los filtros basados en bloques de puntos utiliza las subventanas de la figura 2.6 y consta de las siguientes fases (González-Jiménez 2000):

1. Se calcula la media ponderada en cada subventana de 3×3 puntos.
2. Se sustituye el valor del punto central por la media ponderada más próxima.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & \mathbf{13} & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 1 & 2 & 3 \\ 6 & 7 & 8 \\ 11 & 12 & \mathbf{13} \end{bmatrix}$$

(b)

$$\begin{bmatrix} 3 & 4 & 5 \\ 8 & 9 & 10 \\ \mathbf{13} & 14 & 15 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 7 & 8 & 9 \\ 12 & \mathbf{13} & 14 \\ 17 & 18 & 19 \end{bmatrix}$$

(d)

$$\begin{bmatrix} 11 & 12 & \mathbf{13} \\ 16 & 17 & 18 \\ 21 & 22 & 23 \end{bmatrix}$$

(e)

$$\begin{bmatrix} \mathbf{13} & 14 & 15 \\ 18 & 19 & 20 \\ 23 & 24 & 25 \end{bmatrix}$$

(f)

Figura 2.6: Ventana de (a) 5×5 puntos y (b)-(f) subventanas de 3×3 puntos.

Filtros que minimizan la incertidumbre

Los filtros que minimizan la incertidumbre consideran dos requisitos fundamentales:

- El filtrado de la imagen debe reducir el rango de escalas sobre las cuales los cambios de intensidad se producen. El espectro del filtro debería, por tanto, ser suave y estar aproximadamente limitado en una banda del dominio de la frecuencia.
- Los cambios de intensidad en la imagen son provocados por los cambios de iluminación, los cambios en la orientación o distancia desde el punto de vista del observador de las superficies de los objetos y los cambios producidos por los reflejos de las superficies. Como consecuencia de ello, todos los bordes de la imagen no siguen un mismo patrón general, sino que poseen características locales que los diferencian. Este hecho provoca que la operación de suavización sobre un punto de la imagen deba considerar a los puntos próximos a él y no a puntos que se encuentren muy dispersados. Por tanto, el filtro debería ser suave y localizado en el dominio espacial y, en particular, su varianza espacial (Δx) debería ser pequeña.

Desafortunadamente, estos dos requisitos, uno en el dominio de la frecuencia y otro el dominio espacial, son conflictivos. Están, de hecho, relacionados por el principio de incertidumbre que establece que (Marr y Hildreth 1980)

$$\Delta x \Delta \omega \geq \frac{1}{4} \pi \quad (2.21)$$

donde Δx es la varianza espacial y $\Delta \omega$ es la varianza en el dominio de la frecuencia.

La clase de funciones reales que minimiza esta relación es la clase de funciones gaussianas (figura 2.7):

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.22)$$

donde $\sigma > 0$.

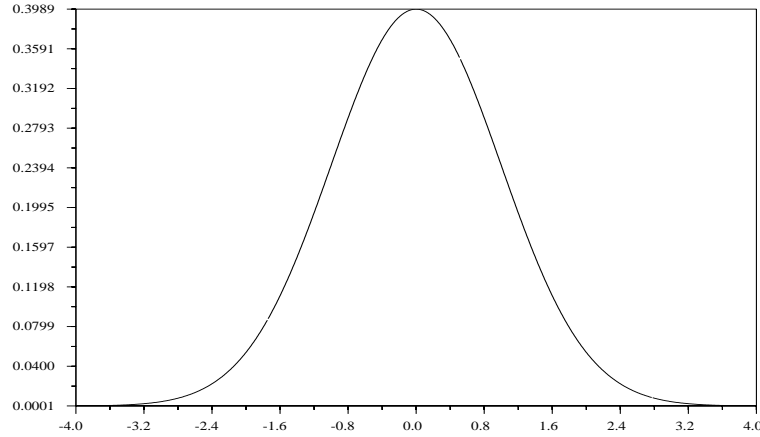


Figura 2.7: Función gaussiana ($\sigma = 1$).

El uso de la función gaussiana como un filtro previo a la diferenciación fue propuesto por Marr y Hildreth (1980). La versión¹ bidimensional de esta clase de funciones es (figura 2.8):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.23)$$

Para suavizar una imagen con un filtro bidimensional gaussiano, se ha de efectuar la siguiente convolución:

$$g(x_0, y_0) = \frac{1}{2\pi\sigma^2} \sum_{i=-n}^n \sum_{j=-n}^n e^{-\frac{i^2+j^2}{2\sigma^2}} f(x_0 + i, y_0 + j) \quad (2.24)$$

Al aplicar el filtro de la gaussiana, ha de tenerse en cuenta que el tamaño de la ventana depende de la desviación típica σ . A medida que aumenta el valor

¹Algunos autores definen la función gaussiana usando diferentes factores de normalización (Sonka et al. 1993)

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

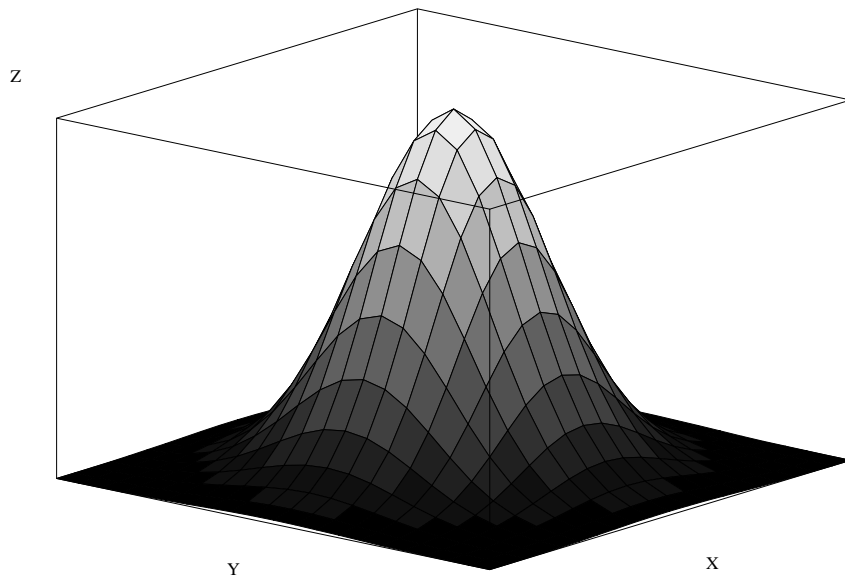


Figura 2.8: Función gaussiana bidimensional ($\sigma = 1$).

de σ , los términos de la función gaussiana alejados de la media (0 en este caso) son mayores y no deben ser despreciados. Una de las relaciones más aceptadas entre el tamaño de la máscara y la desviación típica es

$$N \geq 3c \quad (2.25)$$

donde

$$c = 2\sqrt{2}\sigma \quad (2.26)$$

es el tamaño del lóbulo central de la gaussiana (González-Jiménez 2000).

Esta relación indica que, incluso para valores de σ pequeños, el tamaño de la máscara ha de ser muy grande: para $\sigma = 1$, se obtendría $N \geq 9 > 6\sqrt{2}$. Sin embargo, la función gaussiana es separable por filas y columnas:

$$G(x, y) = G_1(x)G_2(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \quad (2.27)$$

Esta versión del filtro gaussiano se aplicaría en dos etapas: en la primera se realizaría una convolución por filas y al resultado obtenido se le aplicaría una convolución por columnas. La principal ventaja de este método es que el coste computacional se reduce de N^2 a $2N$ operaciones por punto.

Además del filtro basado en la función gaussiana, se han propuesto otros filtros de convolución:

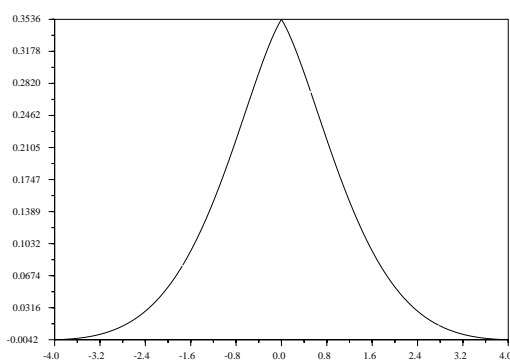
- La función de Green (Ziou y Tabbone 1998):

$$r(x) = \frac{\mu}{2\sqrt{2}} e^{-\mu|x|} \left(\cos\left(\frac{\mu|x|}{\sqrt{2}}\right) + \operatorname{sen}\left(\frac{\mu|x|}{\sqrt{2}}\right) \right) \quad (2.28)$$

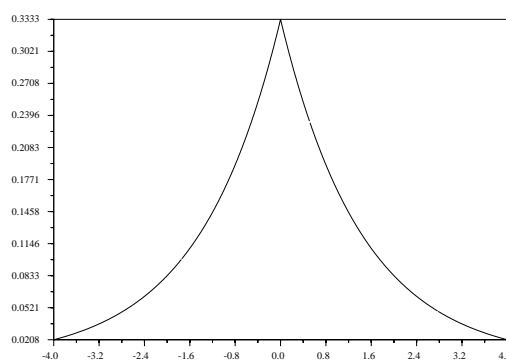
donde $\mu > 0$.

- La función exponencial simétrica (Shen y Castan 1986):

$$h(x) = \frac{a_0}{2 - a_0} (1 - a_0)^{|x|} \quad (2.29)$$



(a)



(b)

Figura 2.9: Filtros de suavización: (a) función de Green ($\mu = 1$) y (b) función exponencial simétrica ($a_0 = 0.5$).

2.5.2.2 Filtros no lineales

Los filtros espaciales no lineales asignan a cada punto de la imagen un valor obtenido mediante la combinación no lineal de los valores de los puntos situados en su entorno. Se han propuesto numerosas técnicas no lineales para suprimir el ruido (Plataniotis y Venetsanopoulos 2000):

- Filtro de la mediana.
- Técnicas basadas en órdenes estadísticos.
- Técnicas basadas en la suavización por aproximación.
- Técnicas homomórficas.
- Técnicas basadas en la morfología matemática.

Filtro de la mediana

El filtro de la mediana pertenece a la clase de los filtros basados en el rango de los valores (*rank value filters*) (Jähne 1993). Estos filtros ordenan los valores pertenecientes a los puntos de un entorno, generalmente una ventana cuadrada de 3×3 , y seleccionan un valor entre ellos: el mínimo, el máximo o la mediana.

El filtro de la mediana reemplaza el nivel de intensidad de cada punto por la mediana de los valores de los puntos en el vecindario de ese punto, ya que es muy improbable que un valor ruidoso llegue a ser la mediana de un conjunto de valores (González y Woods 1996; González-Jiménez 2000). En el caso de una imagen en color, este proceso sería aplicado de forma independiente a cada una de las componentes del espacio de color.

10	8	13
14	92	16
8	12	11

 \Rightarrow

8	8	10	11	12	13	14	16	92
---	---	----	----	-----------	----	----	----	----

 \Rightarrow

10	8	13
14	12	16
8	12	11

Figura 2.10: Filtro de la mediana usando una máscara de 3×3 puntos.

El filtro de la mediana es especialmente efectivo cuando los puntos ruidosos toman valores muy diferentes del resto de los puntos y se desea preservar los bordes de los objetos de la imagen.

El filtro de la mediana suprime el ruido, especialmente el impulsivo, de una manera muy eficiente (figura 2.11). Sin embargo, presenta dos inconvenientes:

- Puede producir el desplazamiento de los bordes (Davis 1990; González-Jiménez 2000)
- Posee un alto coste computacional debido a que han de ordenarse los valores del entorno para poder seleccionar la mediana.

Para abordar el problema del coste computacional, se han propuesto varias alternativas para calcular la mediana de N puntos, como:

- Eliminar los k valores mayores y los k valores menores y promediar el resto (González-Jiménez 2000).
- El método “Ataman” basado en el uso del bit más significativo (Davis 1990).
- El método de “la pseudo - mediana”, basado en el cálculo de máximos y mínimos de subsecuencias (Pratt 1991).

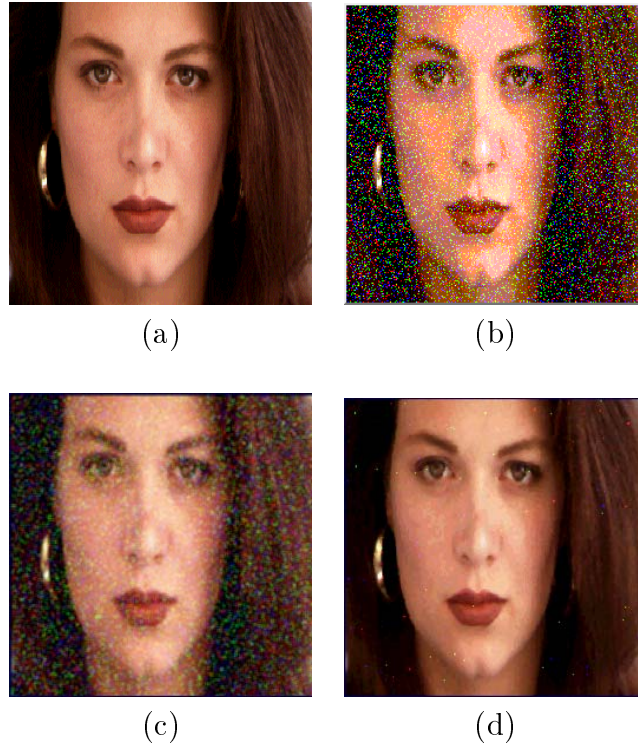


Figura 2.11: (a) Imagen original; (b) imagen degradada con un 10% de ruido impulsivo de “sal y pimienta”; (c) imagen obtenida al aplicar el filtro paso bajo; (d) imagen obtenida al aplicar el filtro de la mediana.

Filtros basados en órdenes estadísticos

Se han propuesto numerosos filtros basados en órdenes estadísticos (Hardie y Arce 1991). Estos filtros utilizan un ordenamiento para identificar a aquellos puntos que no son representativos de la señal y que reciben el nombre de “*outliers*”. Puesto que cada punto ha de ser procesado teniendo en cuenta los valores de los puntos vecinos, se ha de rechazar o atenuar la influencia de los *outliers*.

En el caso de las imágenes en color, estos filtros reciben la denominación de “filtros vectoriales direccionales” (Plataniotis et al. 1997; Trahanias et al. 1996). El filtro del “vector de la mediana” es el representante más conocido de este tipo de filtros (Astola et al. 1988; Astola et al. 1990). La aplicación de este filtro en imágenes definidas en el espacio de color RGB consta de los siguientes pasos.

1. Asignar a cada punto (x, y) de la imagen un vector

$$\vec{f}(x, y) = (r, g, b) \quad (2.30)$$

2. Dado un entorno centrado sobre un punto (x_0, y_0) de la imagen, calcular para cada vector $\vec{f}(x_i, y_i)$ de dicho entorno la suma de las distancias a todos

los demás vectores:

$$S_i = \sum_{j=1}^n ||\vec{f}(x_i, y_i) - \vec{f}(x_j, y_j)||, i = 1, \dots, n \quad (2.31)$$

donde $|| \cdot ||$ es una distancia métrica.

3. Determinar el vector $\vec{f}(x_{min}, y_{min}) = (r_{min}, g_{min}, b_{min})$ tal que

$$S_{min} = \min_{i=1, \dots, n} \{S_i\} \quad (2.32)$$

4. Asignar al punto (x_0, y_0) el valor de $(r_{min}, g_{min}, b_{min})$.

En la mayoría de las ocasiones, los puntos ruidosos o *outliers* nunca ocupan el lugar de (x_{min}, y_{min}) puesto que sus distancias respecto de los demás puntos serán las de mayor magnitud.

El filtro del vector mediana posee propiedades similares a las del filtro de la mediana de la versión escalar, ya que permite eliminar eficientemente el ruido impulsivo. Existen numerosas variantes de este método que también tienen un buen rendimiento ante ruido gaussiano (Platanotis et al. 1997; Trahanias et al. 1996).

Técnicas basadas en la suavización por aproximación

El objetivo que pretende alcanzar la suavización mediante aproximación es obtener una función “suavizada” y con una expresión analítica conocida que se ajuste lo máximo posible a la función que representa a la imagen (Faugeras 1993).

Su formulación unidimensional es la siguiente: si $f(x)$ denota una función continua de intensidad, que ha sido digitalizada en un conjunto discreto de puntos x_k , $1 < k < N$, y $g(x)$ denota la función suavizada que se quiere calcular, entonces, aplicando la teoría de la regularización, $g(x)$ debe minimizar la expresión (Hildreth 1992; Poggio et al. 1988):

$$\sum_{k=1}^N (f(x_k) - g(x_k))^2 + \lambda \int (g''(x))^2 dx \quad (2.33)$$

El primer término mide la adecuación o ajuste de $g(x)$ a $f(x)$ y el segundo término mide la suavidad de $g(x)$. La constante λ controla las compensaciones entre estas dos medidas.

Un primer representante de este tipo de métodos de suavización por aproximación es el operador de Hueckel (Hueckel 1971; Hueckel 1973; Hueckel 1974; Pratt 1978; Rosenfeld y Kak 1982a). Los métodos de ajuste de superficies han

usado una gran variedad de funciones básicas para realizar la aproximación, incluyendo funciones planas (Haralick 1980), funciones cuadráticas (Prewitt 1970; Schetzen 1982) y los polinomios (Cotroulos y Pitas 1994; Haralick 1983b; Haralick 1984; Schetzen 1982). También se ha propuesto un método que usa una gran variedad de superficies direccionales (superficies que son constantes en alguna dirección) para aproximar la función de intensidad bidimensional. La sección unidimensional de estas superficies podía ser lineal, cuadrática, cúbica o la función tangente hiperbólica (Nalwa y Binford 1986).

Los métodos de suavización mediante aproximación están muy relacionados con los métodos de suavización basados en el filtrado de la imagen. Esta afirmación es corroborada por el hecho de que la solución al problema de aproximación de 2.33 es equivalente a la convolución de la imagen con un *spline* cúbico (o sus derivadas) (Faugeras 1993; Poggio et al. 1988; Ziou y Tabbone 1998).

Técnicas homomórficas

Los filtros homomórficos y sus extensiones constituyen una de las primeras clases de filtros no lineales y han sido muy utilizados en el procesamiento de señales y de imágenes digitales. Su característica básica es que usan operaciones no lineales (principalmente el logaritmo) para transformar las señales no lineales en señales aditivas, para utilizar después filtros lineales. El proceso termina transformando la salidas del filtro lineal mediante la operación inversa del método no lineal utilizado (Oppenheim et al. 1968; Pitas et al. 1990).

Técnicas basadas en la morfología matemática

Los filtros morfológicos utilizan las características geométricas de la señal antes que las características analíticas. Este tipo de filtros usan unos operadores morfológicos que pueden ser aplicados a imágenes binarias, monocromáticas o en color. Estos operadores utilizan unas pequeñas figuras geométricas denominadas “elementos estructurantes” para procesar las imágenes. Debido a ello, la morfología matemática también es denominada “morfología estructural” (Cazorla et al. 1997; Dougherty y Astola 1994; González y Woods 1996; Maragos y Schafer 1990; Maragos 1996; Serra 1982).

2.6 Estimación del ruido

El conocimiento previo de la cantidad de ruido que contiene una imagen puede mejorar el rendimiento de los algoritmos de procesamiento digital de imágenes.

Se han estudiado seis métodos de estimación del ruido (Olsen 1993). Tales métodos fueron aplicados a imágenes que contenían ruido aditivo, estacionario y con media cero, es decir, ruido blanco

$$g(x, y) = f(x, y) + \eta(x, y) \quad (2.34)$$

donde $g(x, y)$ es la imagen observada, $f(x, y)$ es la imagen ideal y $\eta(x, y)$ es el ruido. El objetivo que debían cumplir los métodos era estimar la desviación típica σ_η de $\eta(x, y)$.

Los métodos estudiados por Olsen se pueden agrupar en dos clases

- Métodos que filtran la imagen $g(x, y)$ y después calculan σ_η a partir de la imagen filtrada:
 - El filtro de la media o filtro espacial paso bajo.
 - El filtro espacial de la mediana.
 - Método que suaviza la imagen observada $g(x, y)$ y que utiliza el histograma de la magnitud del gradiente (Bracho y Sanderson 1985; Vorhees y Poggio 1987).
- Métodos que utilizan un conjunto de regiones o bloques de la imagen $g(x, y)$ para calcular σ_η
 - Método basado en la media de la varianzas más pequeñas de los bloques de la imagen (Lee 1981).
 - Método basado en la dispersión de σ^2 frente a μ^2 , donde σ y μ son la desviación típica y la media, respectivamente, de los bloques de la imagen seleccionados (Lee y Hoppel 1989).
 - Método que utiliza las varianzas de los bloques para interpolar el valor de σ_η (Meer et al. 1990).

Los métodos fueron aplicados a nueve imágenes, de las cuales cuatro eran artificiales o sintéticas. A las imágenes se les añadieron dos tipos de ruido, ruido gaussiano y ruido uniforme, ambos con media cero. La desviación estándar tomó valores en el conjunto $\{2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. También se estimó el valor de σ_η en las imágenes sin ruido.

Los resultados que obtuvo Olsen fueron algo sorprendentes:

- En media, los métodos basados en un filtramiento previo de la imagen se comportan mejor que los métodos basados en bloques.
- Excepto para valores pequeños de σ_η , tanto el filtro de la media como el de la mediana se comportan mejor que el estadísticamente más sofisticado método de Meer.

- Basados en el porcentaje de éxito acumulado, la conclusión es que el método de la media es el más útil de los seis métodos.
- Sin embargo, los experimentos mostraron que el comportamiento de los métodos depende de las imágenes y del rango de valores σ_η usados.

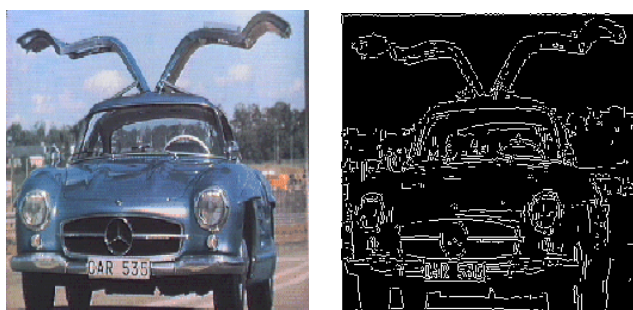
Capítulo 3

Algoritmos de detección de bordes

3.1 Introducción

Los algoritmos de detección de bordes tratan de extraer los objetos de la imagen localizando sus contornos o fronteras (Faugeras 1993; González y Woods 1996; Hildreth 1992; Maravall 1993; Sonka et al. 1993).

Un detector de bordes recibe como entrada una imagen digitalizada y produce como salida un mapa de bordes. El mapa de bordes de algunos detectores incluye información explícita sobre la posición y la fuerza de los bordes, su orientación y escala. La figura 3.1 muestra sólo información posicional.



(a)

(b)

Figura 3.1: (a) Imagen original e (b) información suministrada por un detector de bordes.

La importancia de las técnicas de detección de bordes radica en el hecho de que muchas tareas del procesamiento digital de imágenes (v.g.: realce de imágenes, segmentación de imágenes, compresión de imágenes, reconocimiento

de objetos, reconstrucción 3D, etc.) dependen de la precisión con la que los bordes de los objetos sean obtenidos. Al utilizar una técnica de detección de bordes, se está asumiendo que los bordes contienen una información esencial de la imagen (Hildreth 1992; Zhu et al. 1999; Ziou y Tabbone 1998).

En el procesamiento digital de imágenes, el concepto de “borde” es definido como una discontinuidad en algún atributo de la imagen (Hildreth 1992; Ziou y Tabbone 1998; Zhu et al. 1999). El atributo que ha sido utilizado más frecuentemente es la intensidad luminosa de los niveles de gris de la imagen (González y Woods 1996; Pratt 1978; Rosenfeld y Kak 1982a; Sonka et al. 1993). Sin embargo, un atributo que ha de tenerse muy en cuenta en la detección de bordes es el color. La principal razón se encuentra en el hecho de que una imagen en color suministra más información que una imagen monocromática (Koschan 1995; Nevatia 1977; Novak y Shafer 1987). En una imagen monocromática no se pueden apreciar los bordes existentes entre objetos que tengan diferentes tonos (*hue*) de color y un mismo nivel de intensidad luminosa, ya que serían considerados como partes indistinguibles de un mismo objeto (Koschan 1995; Zhu et al. 1999). Debido a que la capacidad de distinguir los objetos es crucial para muchas aplicaciones de procesamiento digital de imágenes, la información adicional suministrada por el color tiene una gran importancia (Zhu et al. 1999). Además la detección de bordes en color también supera a la detección monocromática de bordes en imágenes con contraste bajo (Koschan 1995; Zhu et al. 1999).

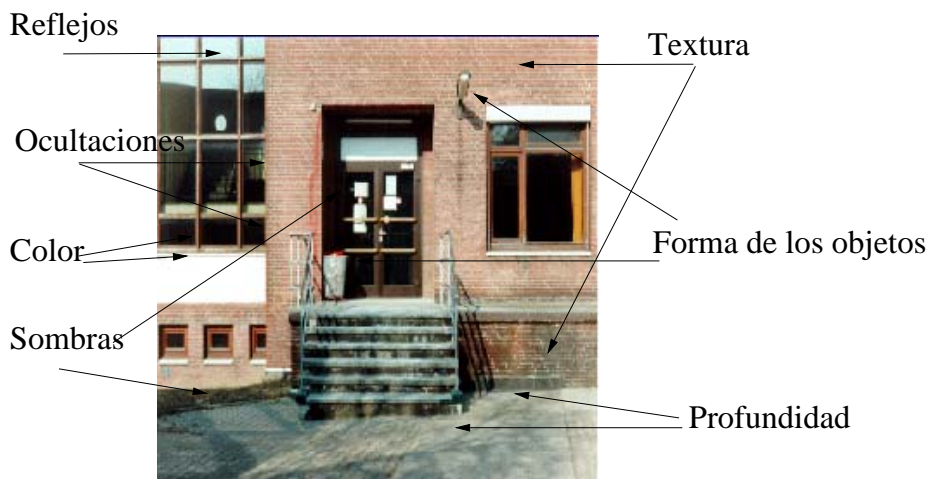


Figura 3.2: Imagen en color con bordes provocados por diferentes motivos.

Los bordes presentes en una imagen pueden ser provocados por muchas causas físicas (Faugeras 1993; Hildreth 1992; Stokman y Gevers 1999; Ziou y Tabbone 1998): iluminación, reflejos, sombras, geometría de los objetos, profundidad de los objetos en la escena, ocultaciones parciales de los objetos, la textura de los objetos, el color, cambios materiales, etc. La figura 3.2 muestra una imagen con bordes provocados por diferentes causas.

3.2 Tipos de bordes

El conocimiento de los diferentes tipos de bordes que pueden aparecer en una imagen es esencial para poder diseñar un buen detector de bordes. En las subsecciones 3.2.1 y 3.2.2 se van a describir, respectivamente, los diferentes tipos de bordes existentes en las imágenes monocromáticas y en las imágenes en color.

3.2.1 Bordes en imágenes monocromáticas

Los bordes de las imágenes monocromáticas se corresponden con los cambios o discontinuidades de la función de intensidad de los niveles de gris. La forma de la función de intensidad a lo largo de una línea de la imagen que sea perpendicular a un borde permite diferenciar los siguientes tipos de bordes (Chidiac y Ziou 1999; Ziou y Tabbone 1998):

- Escalón o salto (*step edge*).
- Rampa (*ramp edge*): puede ser con pendiente cóncava (*concave slope*), con pendiente convexa (*convex slope*) o con ambas.
- Escalera (*staircase edge*).
- Pico (*peak edge*), cresta (*ridge edge*) o pulso (*pulse edge*).
- Valle (*valley edge*).
- Tejado (*roof edge*).

La figura 3.3 muestra los perfiles de la función de intensidad correspondientes a cada uno de los tipos de bordes. Dicho perfiles se obtendrían al tomar los valores de la función de intensidad a lo largo de una línea de la imagen que fuera perpendicular al borde.

La mayoría de los algoritmos de detección de bordes han sido diseñados para detectar bordes del tipo escalón (*step edge*). Este tipo de borde se caracteriza por su ruido, contraste, pendiente y anchura (Zhu et al. 1999). Aparece cuando coinciden dos áreas o regiones homogéneas que poseen niveles de gris muy diferentes entre sí. El borde se sitúa en el punto en el cual la discontinuidad de los niveles de gris se produce. Este tipo de borde se produce cuando hay una sombra sobre un objeto o cuando un objeto oculta parcialmente a otro. La figura 3.4 muestra un modelo ideal del borde de tipo escalón o salto.

Los bordes de tipo pico, cresta o pulso son originados por líneas que atraviesan la imagen. Dichas líneas pueden ser producidas por objetos que estén juntos

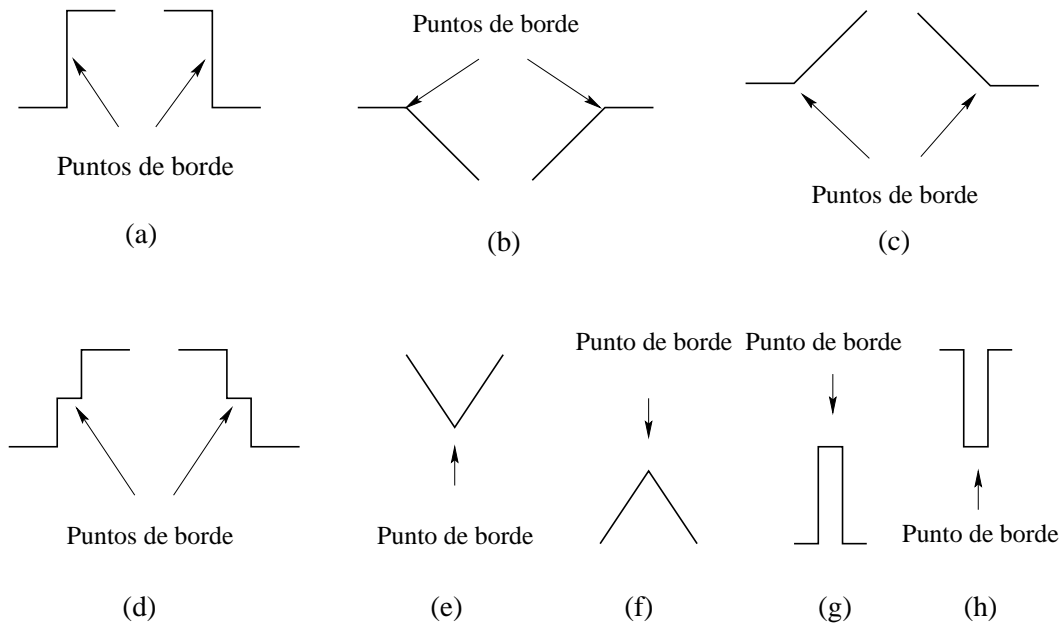


Figura 3.3: Tipos de bordes: (a) escalón o salto, (b) rampa convexa, (c) rampa cóncava, (d) escalera, (e) valle, (f) tejado, (g) y (h) pico o pulso.

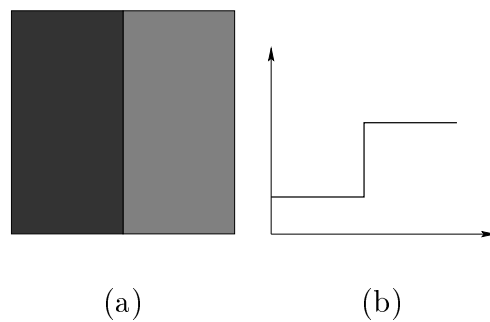


Figura 3.4: Borde ideal de tipo escalón o salto: (a) imagen y (b) perfil de la función de intensidad a lo largo de una línea horizontal de la imagen.

o por objetos delgados situados sobre un fondo. La figura 3.5 muestra una imagen artificial con dos bordes provocados por dos líneas.

Si se analiza la forma geométrica de los bordes, éstos pueden ser clasificados como:

- Líneas rectas que pueden ser horizontales, verticales, diagonales o con cualquier otra orientación.
- Líneas curvas.
- Uniones (*junctions*).
- Esquinas (*corners*).

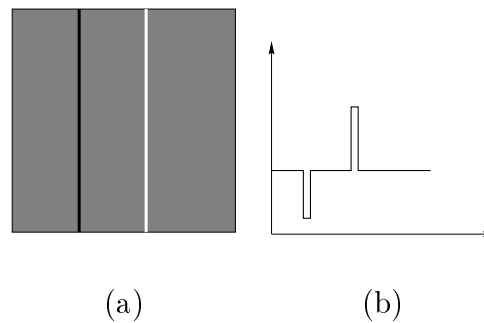


Figura 3.5: Bordes provocados por líneas: (a) imagen y (b) perfil de la función de intensidad a lo largo de una línea horizontal de la imagen.

Las uniones y esquinas se forman cuando al menos dos bordes físicos se encuentran. Hay otras circunstancias adicionales que pueden provocar esquinas en la imagen, por ejemplo, los efectos de la iluminación o la presencia de oclusiones. Hay varios modelos de esquinas o uniones: T , L , Y , X . La figura 3.6 muestra un borde del tipo T-uni3n (T -junction) provocado por la concurrencia de bordes de tipo escal3n.

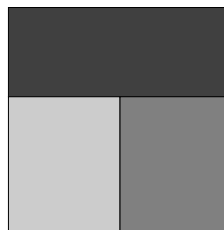


Figura 3.6: Imagen con un borde del tipo T-uni3n.

La influencia de las caracter3sticas de los bordes en el funcionamiento de los detectores ha sido estudiada por muchos investigadores (Abdou 1978; Kitchen y Rosenfeld 1981; Kitchen y Malin 1989; Lyvers y Mitchell 1988; Ponce y Brady 1985; Michelli et al. 1989; Venkatesh y Kitchen 1992; Ziou y Tabbone 1998).

3.2.2 Bordes en im3genes en color

Un borde de una imagen en color suele ser definido como una discontinuidad en el espacio tridimensional de color. Estas discontinuidades pueden ser producidas por cambios en una o en varias de las componentes del espacio de color (Krishnamoorthi y Bhattacharyya 1997; Trahanias y Venetsanopoulos 1996; Yang y Tsai 1996).

Se han propuesto diferentes clasificaciones de bordes en color:

- Clasificaci3n de los bordes seg3n su origen (Stokman y Gevers 1999; Stok-

man y Gevers 2000a; Stokman y Gevers 2000b).

- Clasificación de los bordes según los cambios en un campo vectorial (Plataniotis y Venetsanopoulos 2000).
- Clasificación de los bordes según la información de color de su alrededor (Nakamura et al. 1998).

Clasificación de los bordes en color según su origen

Se han clasificado los bordes en color dependiendo de su origen (Stokman y Gevers 1999; Stokman y Gevers 2000a; Stokman y Gevers 2000b):

- Bordes provocados por formas geométricas.
- Bordes provocados por reflejos.
- Bordes provocados por sombras
- Bordes provocados por cambios materiales.

Para determinar el origen de un borde en color, se han propuesto tres métodos que analizan la sensibilidad o invarianza de los módulos de los gradientes calculados en tres espacios de color diferentes. Los métodos se diferencian en los espacios de color utilizados:

- La tabla 3.1 muestra la clasificación de los bordes basada en los módulos de los gradientes calculados en un espacio de color hiperespectral ($||\vec{\nabla}_{raw}||$), en un espacio de color hiperespectral normalizado ($||\vec{\nabla}_{norm}||$) y en la componente de tono (*hue*) del espacio de color hiperespectral ($||\vec{\nabla}_{hue}||$) (Stokman y Gevers 1999)

Tabla 3.1: Clasificación de los bordes propuesta en (Stokman y Gevers 1999) (“+” denota sensibilidad; “-”, invarianza).

	geometría - sombra	reflejo	material
$ \vec{\nabla}_{raw} $	+	+	+
$ \vec{\nabla}_{norm} $	-	+	+
$ \vec{\nabla}_{hue} $	-	-	+

- La tabla 3.2 muestra la clasificación de los bordes basada en los módulos de los gradientes calculados en los espacios de color RGB , $c_1c_2c_3$ y $l_1l_2l_3$, donde (Stokman y Gevers 2000a)

$$c_1 = \arctan\left(\frac{R}{\max\{G, B\}}\right) \quad (3.1)$$

$$c_2 = \arctan\left(\frac{G}{\max\{R, B\}}\right) \quad (3.2)$$

$$c_3 = \arctan\left(\frac{B}{\max\{R, G\}}\right) \quad (3.3)$$

y

$$l_1 = \frac{|R - G|}{|R - G| + |R - B| + |G - B|} \quad (3.4)$$

$$l_2 = \frac{|R - B|}{|R - G| + |R - B| + |G - B|} \quad (3.5)$$

$$l_3 = \frac{|G - B|}{|R - G| + |R - B| + |G - B|} \quad (3.6)$$

Tabla 3.2: Clasificación de los bordes propuesta en (Stokman y Gevers 2000a) (“+” denota sensibilidad; “-”, invarianza).

	geometría - sombra	reflejo	material
$ \vec{\nabla}_{RGB} $	+	+	+
$ \vec{\nabla}_{c_1c_2c_3} $	-	+	+
$ \vec{\nabla}_{l_1l_2l_3} $	-	-	+

- La tabla 3.3 muestra la clasificación de los bordes basada en los módulos de los gradientes calculados en los espacios de color RGB , rgb y opp , donde opp representa el espacio de color de “colores contrarios” (*opponent color space*) que es definido como (Stokman y Gevers 2000b)

$$o = \frac{R - G}{2} \quad (3.7)$$

$$p = \frac{2B - R - G}{4} \quad (3.8)$$

Tabla 3.3: Clasificación de los bordes propuesta en (Stokman y Gevers 2000b) (“+” denota sensibilidad; “-”, invarianza).

	geometría - sombra	reflejo	material
$ \vec{\nabla}_{RGB} $	+	+	+
$ \vec{\nabla}_{rgb} $	-	+	+
$ \vec{\nabla}_{opp} $	+	-	+

Clasificación de los bordes según los cambios en un espacio vectorial

Si una imagen en color es considerada como un espacio vectorial tridimensional, un borde de color puede ser definido como una discontinuidad significativa en el campo vectorial que representa la función de la imagen en color. Un borde de tipo escalón se caracterizaría por un cambio abrupto en el campo vectorial, mientras que un cambio gradual se correspondería con un borde de rampa. Estas definiciones no pretenden ser definiciones formales, sino más bien descripciones intuitivas de la noción de borde en una imagen en color (Plataniotis y Venetsanopoulos 2000).

Clasificación de los bordes según la información de color de su alrededor

Los bordes de color han sido clasificados en función de la información de color existente a su alrededor (Nakamura et al. 1998). Este método identifica el perfil de cada borde según las características de color en el espacio *RGB* que poseen los dos lados contiguos al borde.

Inicialmente, se caracterizan los modelos o fenómenos físicos que provocan los cambios de intensidad (figura 3.7):

- *Albedo*: se produce un cambio través del borde pero no existe ninguna discontinuidad sustancial de profundidad.
- Oclusión (*occlusion*): dos regiones que pertenecen a objetos diferentes se encuentran en el borde. No existe ninguna sombra significativa entre los dos objetos.
- Sombra (*shadow*): dos regiones pertenecen a una región homogénea de un objeto. Una de las regiones se corresponde con la sombra proyectada por otro objeto y la otra región está directamente iluminada por una fuente de luz normal.
- Cresta (*ridge*): dos regiones se juntan en una “cresta” o “valle” de un objeto.

- Compuesto (*compound*): dos objetos se solapan, proyectando uno de ellos su sombra sobre el otro.
- Contacto (*touch*): dos regiones pertenecen a dos objetos que se tocan o están muy próximos. La iluminación es atenuada en el hueco que separa a los objetos.

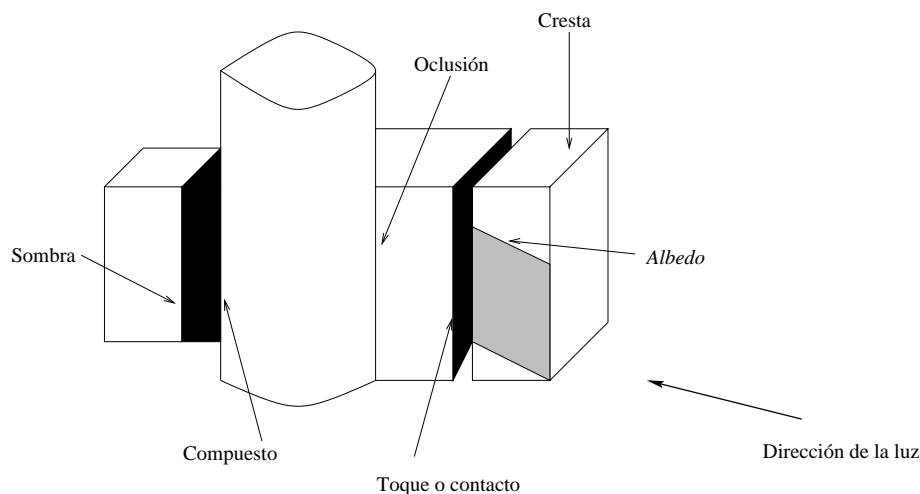


Figura 3.7: Bordes provocados por fenómenos físicos.

Estos modelos o fenómenos no son siempre excluyentes, ya que, por ejemplo, los fenómenos provocados por un “contacto” y una “composición” se pueden combinar.

La caracterización de estos modelos en una imagen en color definida en el espacio de color *RGB* se puede realizar analizando la información de color de las regiones que forman el borde. Si se considera que C_1 y C_2 son los centros de los *clusters* correspondientes a las regiones situadas a ambos lados del borde, los fenómenos anteriores pueden ser caracterizados como:

- *Albedo*: se produce una transición rápida de un *cluster* a otro. El valor p de un punto puede ser modelado como:

$$p = kC_1 + (1 - k)C_2, \quad 0 \leq k \leq 1 \quad (3.9)$$

- *Oclusión*: la caracterización es similar a la del fenómeno de *albedo* salvo que el lado ocultado presenta a menudo una degradación suave (*smooth shading*).
- *Sombra*: los dos centros de los *clusters* y el origen del espacio de color *RGB* están alineados, puesto que se asume que la luz ambiental puede ser modelada como la atenuación de la luz directa.

$$C_1 \cong kC_2, \quad p = k'C_1, \quad k \leq k' \leq 1 \quad (3.10)$$

- Cresta: su caracterización es similar a la de “sombra” pero el borde suele estar acompañado por reflejos.
- Compuesto: los valores de los puntos situados a ambos lados del borde pueden ser caracterizados de una de las dos siguientes formas
 - (a) El valor del punto varía rápidamente desde el valor C_1 , correspondiente a la parte oculta, hasta el valor kC_2 , correspondiente a la parte sombreada del otro lado, es decir, el valor atenuado de C_2 .
 - (b) El valor del punto varía entre el valor atenuado kC_2 y C_2 .

Por tanto, el valor p de un punto es modelado como

$$p = \begin{cases} kC_2 - C_1 & \text{En el caso (a)} \\ k'C_2 & \text{En el caso (b)} \end{cases} \quad 0 \leq k \leq k' \leq 1 \quad (3.11)$$

- Contacto: es similar al caso anterior, pero la parte oculta es pequeña y se puede observar una reflexión mutua.

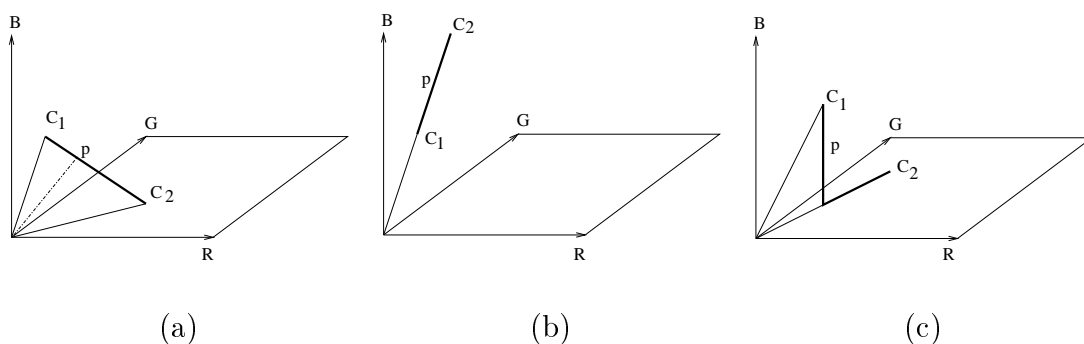


Figura 3.8: Perfiles de los modelos de bordes en el espacio de color RGB : (a) *albedo* - *oclusión*, (b) *sombra* - *cresta* y (c) *compuesto* - *contacto*.

3.3 Detección de bordes en imágenes monocromáticas

3.3.1 Algoritmos de detección de bordes monocromáticos

Los algoritmos de detección de bordes en imágenes monocromáticas han proliferado desde la aparición del procesamiento digital de imágenes. Si se consideran los detectores de bordes como un elemento integrado en un sistema de visión, se pueden clasificar en dos grandes grupos (Ziou y Tabbone 1998):

- Detectores “autónomos”: no utilizan ningún conocimiento a priori, es decir, no están influenciados por otros componentes del sistema de visión ni por información contextual. Suelen realizar un procesamiento local, de forma que los bordes son identificados mediante el análisis de los puntos de su entorno. Estos detectores son flexibles, ya que no están limitados a imágenes específicas y son apropiados para sistemas de visión de propósito general.
- Detectores “contextuales”: utilizan un conocimiento a priori del borde o la escena que van a procesar. Estos métodos dependen de los resultados de otros componentes del sistema de visión. Su funcionamiento está limitado a un contexto preciso donde las imágenes procesadas siempre incluyen los mismos objetos.

Los detectores de bordes contextuales que han sido propuestos son muy pocos en comparación con los autónomos. Una recopilación de los detectores contextuales se puede encontrar en (Davis 1975) y (Matsuyama 1989).

La mayoría de los algoritmos de detección, tanto autónomos como contextuales, utilizan las tres fases fundamentales de suavización, diferenciación e identificación. Se diferencian en los filtros de suavización, los operadores de diferenciación, los procesos de identificación, los objetivos, la complejidad computacional y los modelos matemáticos usados en su diseño.

Los detectores autónomos realizan un procesamiento local para la extracción de los bordes, lo cual llevó a Davis a clasificar las técnicas de detección de bordes en dos clases (Davis 1975):

- Técnicas secuenciales: la decisión para determinar si un punto pertenece o no a un borde depende de los resultados obtenidos por el detector en algunos puntos examinados previamente. El funcionamiento de los métodos secuenciales depende de la elección del punto inicial apropiado y del proceso de selección del siguiente punto que vaya a ser procesado en cada iteración.
- Técnicas paralelas: la decisión para determinar si un punto pertenece o no a un borde depende de dicho punto y de su entorno. Como consecuencia de ello, los operadores de esta clase pueden ser aplicados simultáneamente a todos puntos de la imagen.

Los mayoría de los algoritmos de detección de bordes están basados en:

- Las derivadas de primer orden (v.g.: el gradiente).
- Las derivadas de segundo orden (v.g.: el laplaciano).
- Criterios de optimización.

Sin embargo, también se han utilizado otros modelos matemáticos para el diseño de algoritmos de detección de bordes:

- Ajuste paramétrico (Deriche y Blaszk 1993; Haralick y Shapiro 1985; Haralick 1984; Hueckel 1971; Hueckel 1973; Hueckel 1974; Nalwa y Binford 1986; Nalwa 1987; Nayar et al. 1996; Tabatai y Mitchell 1984).
- Morfología matemática (Bousquet y Flouzat 1986; Hewitt et al. 1990; Krishnapuram y Gupta 1992; Lee et al. 1987; Meyer et al. 1997; Song 1993).
- Análisis de texturas (Barranco-López 1998; Haddon y Boyce 1993; He y Wang 1992; Khotanzad y Chen 1989; Oliver y Lombardo 1996; Xiaohan et al. 1991; Yu et al. 1991).
- Teoría de conjuntos borrosos (Goetcherian 1980; Nakagawa y Rosenfeld 1978; Pal y King 1983).
- Relajación (Deng y Iyengar 1996; Iyengar y Deng 1995; Surka y Valavanis 1991; Zucker et al. 1977).
- Redes neuronales (Pinho y Almeida 1997; Shah 1990; Srinivasan et al. 1994; Varona y Villanueva 1997).
- Algoritmos genéticos (Bhandarkar et al. 1994).

Además también se han diseñado algoritmos específicos para la detección de líneas (Burns y Hanson 1986; Fischler et al. 1981; Giraudon 1985; Haralick 1983a; Hueckel 1973; Hueckel 1974; Koundinya y Chanda 1994; Ziou 1991), uniones y esquinas (Cooper et al. 1993; Deriche y Blaszk 1993; Guiducci 1988; Huang y Yan 1995; Kholmann 1996; Lindeberg 1994; Mehrotra et al. 1990; Noble 1992; Rangarajan et al. 1989; Rattarangsi y Chin 1992; Rohr 1992; Teuber 1992; Tabbone 1994a).

Davis (1975) realizó una revisión de los primeros detectores de bordes. Después han sido varios los estudios o recopilaciones que se han publicado sobre este tema (Nalwa 1993; Pal y Pal 1993; Torre y Poggio 1986; Zamperoni 1995; Ziou y Tabbone 1998). Además, la detección de bordes ha sido abordada en diferentes textos sobre procesamiento digital de imágenes (Faugeras 1993; González y Woods 1996; Levine 1985; Pitas 1993; Pratt 1978; Rosenfeld y Kak 1982a; Sonka et al. 1993).

3.3.2 Fases en la detección de bordes en imágenes monocromáticas

La mayoría de los algoritmos de detección de bordes en imágenes monocromáticas han sido diseñados para bordes de tipo salto o escalón e incluyen tres operaciones

fundamentales (Hildreth 1992; Ziou y Tabbone 1998):

- Suavización de la imagen: trata de reducir el ruido de la imagen y asegurar una correcta detección de los bordes.
- Diferenciación de la imagen: evalúa las derivadas de la imagen para resaltar las características de los bordes.
- Identificación de los bordes: pretende localizar los bordes reales y suprimir los falsos bordes.

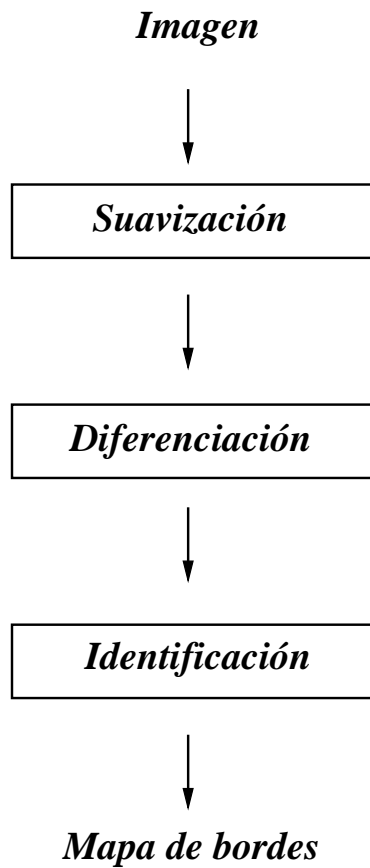


Figura 3.9: Fases de la detección de bordes.

La operación de identificación siempre debe ser la última en ejecutarse. Sin embargo, las operaciones de diferenciación y suavización pueden permutarse si se utilizan operadores lineales para llevarlas a cabo. Ello es debido a que los operadores lineales son asociativos y conmutativos con la convolución, es decir:

$$\mathcal{O}_{x,y}(f(x,y) * h(x,y)) = \mathcal{O}_{x,y}(f(x,y)) * h(x,y) = f(x,y) * \mathcal{O}_{x,y}(h(x,y)) \quad (3.12)$$

donde $f(x,y)$ y $h(x,y)$ han de ser dos funciones de clase C^2 según el operador lineal $\mathcal{O}_{x,y}$.

Si los operadores utilizados para suavizar y diferenciar la imagen no son lineales, entonces las operaciones no son conmutativas y la suavización debe preceder a la operación de diferenciación (Ziou y Tabbone 1998).

La correcta identificación de los bordes es un problema muy importante, porque la naturaleza discreta de las imágenes digitales puede provocar un desplazamiento del borde real de la localización entera más próxima¹.

Estas tres operaciones están estrechamente relacionadas, ya que la suavización regulariza la diferenciación y la identificación de los bordes depende del funcionamiento de las otras dos operaciones. Si la etapa de suavización reduce el ruido sin pérdida de información, la supresión de bordes falsos se puede hacer más fácilmente.

La especificación de un detector de bordes en términos de estas tres operaciones es incompleta, porque no se incluye ninguna información precisa del contexto en el cual puede ser utilizada mejor, ni información sobre la escala de computación. Además, hay otros tipos de bordes, como las líneas, uniones y esquinas que requieren métodos de detección diferentes. Se necesita, por tanto, definir una metodología de detección de bordes para hacer explícito cómo se va a elegir la escala (“multiescala”) y cómo se va a seleccionar un detector de bordes (“multidetector”) para una aplicación concreta.

En los siguientes apartados se van a describir las tres operaciones fundamentales de la detección de bordes y los métodos para la detección de puntos, líneas y uniones. Finalmente, se justificara el uso de los métodos de detección de bordes con múltiples escalas y los multidetectores.

3.3.3 Suavización de la imagen

La suavización de la imagen presenta las siguientes características (Hildreth 1992; Ziou y Tabbone 1998):

- Efectos positivos: reduce el ruido presente en la imagen y asegura una robusta detección de los bordes.
- Efectos negativos: pérdida de información.

Para que la operación de diferenciación sea más robusta frente al ruido, debe realizarse una suavización de la imagen. Al suavizar la imagen, se establece la resolución o escala en la que se producirá la detección de los bordes. La correcta elección de dicha resolución plantea un importante dilema entre la pérdida de información y la reducción de ruido. El objetivo final es encontrar el detector

¹Este problema suele ser citado como “*subpixel resolution*”.

de bordes que asegure un compromiso favorable entre la reducción de ruido y la conservación del borde.

Se dice que un problema matemático está “bien planteado” si (Faugeras 1993; Hadamard 1923):

- Tiene solución.
- La solución es única.
- Es robusto frente al ruido.

La diferenciación numérica de la imagen es “un problema mal planteado”, porque su solución no es robusta frente al ruido. La operación de suavización sirve para regularizar la imagen, provocando que la operación de diferenciación esté bien planteada (Torre y Poggio 1986).

Los métodos de suavización más utilizados por los detectores de bordes son²

- Filtros espaciales lineales, especialmente los filtros de “paso bajo” y el filtro de la gaussiana.
- Filtros de suavización mediante aproximación.

3.3.4 Diferenciación de la imagen

Los bordes se corresponden con zonas de la imagen en las que se producen discontinuidades o cambios bruscos en el nivel de intensidad. Estos cambios de intensidad pueden ser acentuados mediante operaciones de diferenciación. Al realizar la diferenciación, se obtiene una representación de la imagen que facilita la extracción de las propiedades de los bordes. Los métodos más usados para realizar la diferenciación son los basados en las primeras derivadas, ya que éstas miden la variación de los valores de la función. Para detectar los cambios bruscos de intensidad en una imagen se suelen utilizar dos aproximaciones:

- Localización de los extremos (máximos o mínimos) de la primera derivada de la función de intensidad.
- Localización de los cruces por cero (*zero crossings*) o transiciones de valores negativos a positivos, o viceversa, de la segunda derivada de la función de intensidad.

²Estos filtros de suavización son descritos en la sección 2.5.

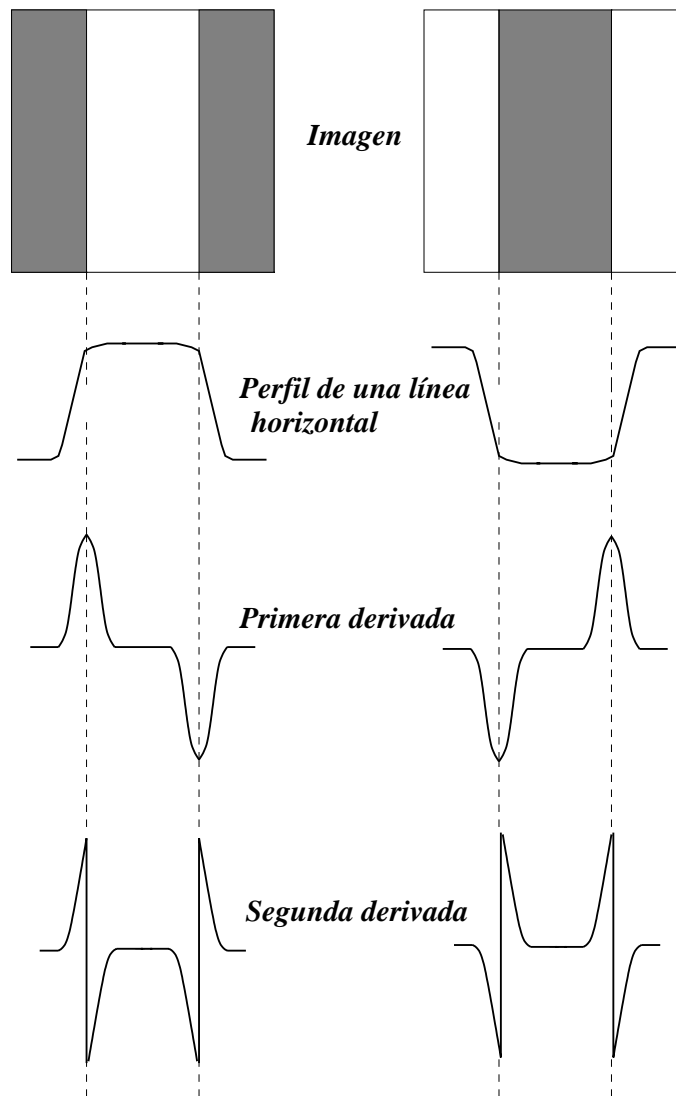


Figura 3.10: Comportamiento de las derivadas de primer y segundo orden ante la presencia de un borde (González y Woods 1996).

Considérese la figura 3.10, en la cual se muestran dos imágenes con bordes de tipo escalón o salto, el perfil de la función de intensidad y sus derivadas de primer y segundo orden. A la izquierda se muestra una franja blanca sobre un fondo oscuro, mientras que en la parte derecha se muestra una franja oscura sobre un fondo blanco. De ambas imágenes se ha extraído el nivel de intensidad de los puntos situados a lo largo de una línea horizontal.

La primera derivada toma valores positivos cuando hay una transición desde una zona a otra con una intensidad mayor, es decir, más clara. Cuando la transición es hacia zonas con menor intensidad, la derivada toma valores negativos, y cuando se recorre una zona con niveles de intensidad constante, la derivada toma un valor nulo.

Por lo tanto, atendiendo a la magnitud de la primera derivada, se puede detectar la presencia de un borde. La primera derivada siempre presenta un extremo, máximo o mínimo, en los puntos situados exactamente en un borde.

En el caso de la segunda derivada, cuando se produce una transición entre zonas con diferentes niveles de intensidad, toma valores positivos en la zona de la transición asociada a la intensidad menor, y negativos en la zona asociada a la intensidad mayor. Cuando la intensidad permanece constante, la segunda derivada toma un valor nulo. Para determinar si un punto pertenece a la parte oscura o clara de un borde, es suficiente con fijarse en el signo de la segunda derivada. También hay que destacar el hecho de que en un punto situado exactamente en el borde se produce un cruce por cero (*zero-crossing*), es decir, la función corta el eje de abscisas en dicho punto, pasando de un valor positivo a otro negativo, o viceversa.

Esta descripción se pueden extender al caso bidimensional usando razonamientos análogos. Sólo hay que definir un perfil perpendicular a la dirección del borde en cualquier punto que se considere e interpretar los resultados de forma similar.

La primera derivada en un punto de la imagen es obtenida usando la magnitud del gradiente, mientras que la segunda derivada es obtenida usando el laplaciano.

Además de los detectores de bordes basados en el gradiente o en el laplaciano, se han propuesto otros detectores de bordes que han sido diseñados mediante la optimización de ciertos criterios relacionados con la detección de bordes. El más popular es el algoritmo de Canny (1.986), pero no es el único (Deriche 1987; Shen y Castan 1986; Spacek 1986).

Aunque la mayoría de los detectores de bordes se han centrado en los bordes de tipo escalón o salto, hay que destacar que los bordes de tipo línea han de ser detectados de una manera diferente, ya que se corresponden con los extremos (máximos o mínimos) locales de la imagen. Se localizan en los cruces por cero de la primera derivada o en los máximos locales del laplaciano.

3.3.4.1 Gradiente

El gradiente de una función escalar bidimensional $f(x, y)$ en un punto de coordenadas (x, y) es un vector $\vec{\nabla}f(x, y)$, cuya dirección coincide con la dirección de máxima variación de la función en dicho punto, y cuya magnitud indica el valor de dicha variación por unidad de distancia y en la dirección mencionada. La expresión del vector gradiente es:

$$\vec{\nabla}f(x, y) = (G_x(x, y), G_y(x, y)) = \left(\frac{\partial f}{\partial x}(x, y), \frac{\partial f}{\partial y}(x, y) \right) \quad (3.13)$$

Su magnitud $||\vec{\nabla}f(x, y)||$ y su dirección $\alpha(x, y)$ vienen dadas por las siguientes expresiones:

$$||\vec{\nabla}f(x, y)|| = \sqrt{G_x^2(x, y) + G_y^2(x, y)} \quad (3.14)$$

$$\alpha(x, y) = \arctan\left(\frac{G_y(x, y)}{G_x(x, y)}\right) \quad (3.15)$$

Los puntos de la imagen que pertenecen a los bordes son aquéllos que dan un valor máximo en la magnitud del gradiente.

La dirección del gradiente es perpendicular a la orientación del borde. En muchos métodos propuestos, la dirección del gradiente es usada para localizar los bordes.

El módulo del gradiente en un operador invariante a rotaciones y no lineal. Este módulo es calculado usando sólo las derivadas en x e y . En una imagen ruidosa el uso de varias derivadas direccionales puede ser útil para incrementar la proporción entre la señal y el ruido (*signal to noise ratio*).

Debido a la ineficiencia computacional de la operación de la raíz cuadrada requerida en 3.14, el módulo del gradiente puede ser aproximado mediante

$$||\vec{\nabla}f(x, y)|| \approx |G_x(x, y)| + |G_y(x, y)| \quad (3.16)$$

o

$$||\vec{\nabla}f(x, y)|| \approx \text{máximo}(|G_x(x, y)|, |G_y(x, y)|) \quad (3.17)$$

Estas aproximaciones pueden ser calculadas más eficientemente, pero, en el caso de los bordes de tipo escalón o salto (*step edge*), se reduce la precisión del módulo gradiente estimado (Lyvers y Mitchell 1988).

Son numerosos los métodos que se han utilizado para calcular el vector gradiente en un punto de una imagen digital (figura 3.11), como los operadores de Roberts, Prewitt, Sobel, Kirsch, Robinson, el operador isotrópico o los operadores de “brújula” (*compass operator*) (González y Woods 1996; Jain 1989; Kirsch 1971; Park y Choi 1989; Prewitt 1970; Roberts 1965; Robinson 1977b; Rosenfeld y Kak 1982a; Sobel 1978; Sonka et al. 1993).

3.3.4.2 Derivadas de segundo orden

Los operadores de segundo orden son definidos en términos de $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial x \partial y}$ y $\frac{\partial^2 f}{\partial y^2}$. Los operadores de segundo orden más utilizados son el operador laplaciano y las

$$\begin{array}{ccc}
\begin{bmatrix} \boxed{0} & 1 \\ -1 & 0 \end{bmatrix} & \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix} & \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{bmatrix} \\
\text{(a)} & \text{(b)} & \text{(c)} \\
\frac{1}{5} \begin{bmatrix} -1 & -1 & -1 \\ 1 & \boxed{-2} & 1 \\ 1 & 1 & 1 \end{bmatrix} & \frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ -3 & \boxed{0} & -3 \\ 5 & 5 & 5 \end{bmatrix} & \frac{1}{2+\sqrt{2}} \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & \boxed{0} & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \\
\text{(d)} & \text{(e)} & \text{(f)}
\end{array}$$

Figura 3.11: Máscaras de los operadores de (a) Roberts, (b) Prewitt, (c) Sobel, (d) Robinson, (e) Kirsch y (f) del operador isotrópico para calcular la aproximación a G_y .

derivadas direccionales de segundo orden en la dirección del gradiente. Estos operadores son definidos por:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2}(x, y) + \frac{\partial^2 f}{\partial y^2}(x, y) \quad (3.18)$$

y

$$\frac{\partial^2 f}{\partial \vec{n}^2}(x, y) = \frac{\partial^2 f}{\partial x^2}(x, y)\cos^2(\alpha) + \frac{\partial^2 f}{\partial x \partial y}(x, y)\sin(\alpha)\cos(\alpha) + \frac{\partial^2 f}{\partial y^2}(x, y)\sin^2(\alpha) \quad (3.19)$$

donde \vec{n} es la dirección del gradiente.

Los puntos de la imagen que pertenecen a los bordes son aquéllos en los que el operador de segundo orden produce un cruce por cero (*zero crossings*).

La técnica tradicional de localizar los bordes en los cruces por cero de las derivadas de segundo orden tienen los siguientes inconvenientes (Sonka et al. 1993):

- Suavizan demasiado la forma de la imagen. Por ejemplo, las esquinas nítidas se suelen perder.
- Tienden a crear bordes con lazos cerrados (efecto de “plato de espagueti”).

El operador laplaciano es lineal y rotacionalmente simétrico, mientras que la segunda derivada direccional no es lineal ni invariante a la rotación.

En el análisis digital de imágenes, el cálculo del operador laplaciano se suele efectuar de dos maneras (figura 3.12) (Sonka et al. 1993):

- Utilizando la conectividad 4.

$$\nabla^2 f(x, y) \equiv f(x+1, y) + f(x-1, y) + f(x, y-1) + f(x, y+1) - 4f(x, y) \quad (3.20)$$

- Utilizando la conectividad 8.

$$\begin{aligned} \nabla^2 f(x, y) \equiv & f(x+1, y) + f(x-1, y) + f(x, y-1) \\ & + f(x, y+1) + f(x+1, y+1) + f(x+1, y-1) \\ & + f(x-1, y+1) + f(x-1, y-1) - 8f(x, y) \end{aligned} \quad (3.21)$$

$$\begin{array}{ccc} \left[\begin{array}{ccc} 0 & 1 & 0 \\ 1 & \boxed{-4} & 1 \\ 0 & 1 & 0 \end{array} \right] & & \left[\begin{array}{ccc} 1 & 1 & 1 \\ 1 & \boxed{-8} & 1 \\ 1 & 1 & 1 \end{array} \right] \\ \text{(a)} & & \text{(b)} \end{array}$$

Figura 3.12: Máscaras del operador laplaciano: (a) conectividad 4; (b) conectividad 8.

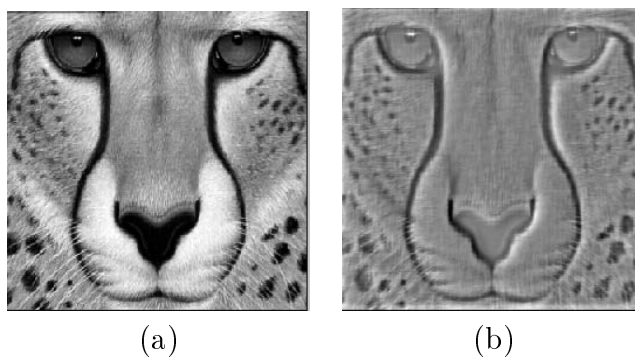


Figura 3.13: (a) Imagen original; (b) imagen obtenida al aplicar el operador laplaciano con conectividad 4.

Es preferible utilizar el operador laplaciano cuando las transiciones de intensidad de la imagen no son muy bruscas. Este operador tiene varios inconvenientes (González y Woods 1996):

- Es demasiado sensible al ruido, lo cual es una característica propia de los operadores de derivadas de segundo orden.
- Produce respuestas dobles para un mismo borde.
- No permite calcular la dirección del borde.

Debido a estas razones, el operador laplaciano tiene generalmente un papel secundario como detector de bordes. Sin embargo, su comportamiento mejora considerablemente si se combina con una operación de suavización que utilice la

función gaussiana (González y Woods 1996). El operador obtenido se denomina “laplaciano de la función gaussiana (LoG)” (Marr y Hildreth 1980). La aplicación de este operador a una imagen $f(x, y)$ origina la siguiente imagen transformada

$$\nabla^2(G(x, y) * f(x, y)) \quad (3.22)$$

El orden de ejecución de la diferenciación y la convolución puede ser intercambiado gracias a la linealidad de los operadores, obteniéndose

$$(\nabla^2 G(x, y)) * f(x, y) \quad (3.23)$$

Puesto que el laplaciano de la función gaussiana es independiente de la imagen, puede ser computado previamente, reduciéndose la complejidad de la operación de composición.

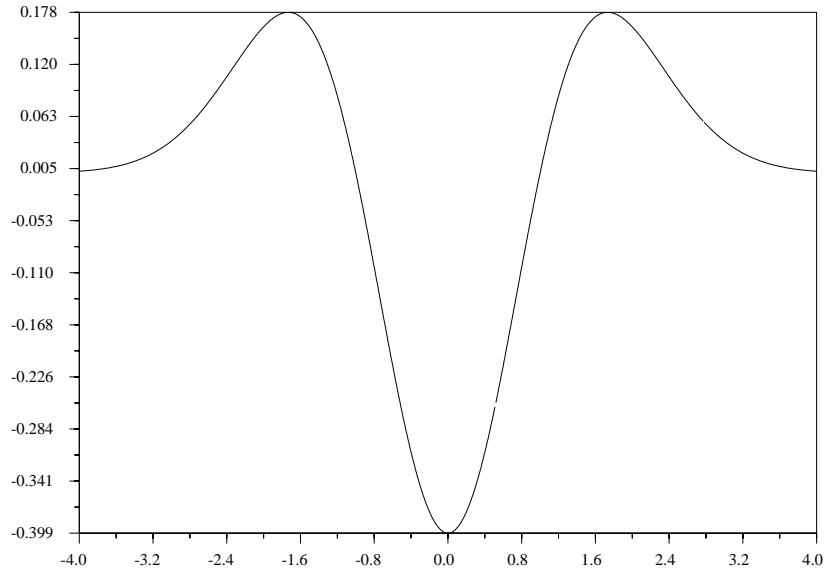


Figura 3.14: Laplaciano de la función gaussiana ($\sigma = 1$).

La versión unidimensional del laplaciano de la función gaussiana (figura 3.14) se define como (Marr y Hildreth 1980) :

$$\nabla^2 G(x) = \frac{-1}{\sqrt{2\pi}\sigma^3} \left(1 - \frac{x^2}{\sigma^2}\right) e^{\left(-\frac{x^2}{2\sigma^2}\right)} \quad (3.24)$$

Análogamente, la versión bidimensional del laplaciano de la función gaussiana se define como

$$\nabla^2 G(x, y) = \frac{-1}{2\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{\sigma^2}\right) e^{\left(-\frac{x^2 + y^2}{2\sigma^2}\right)} \quad (3.25)$$

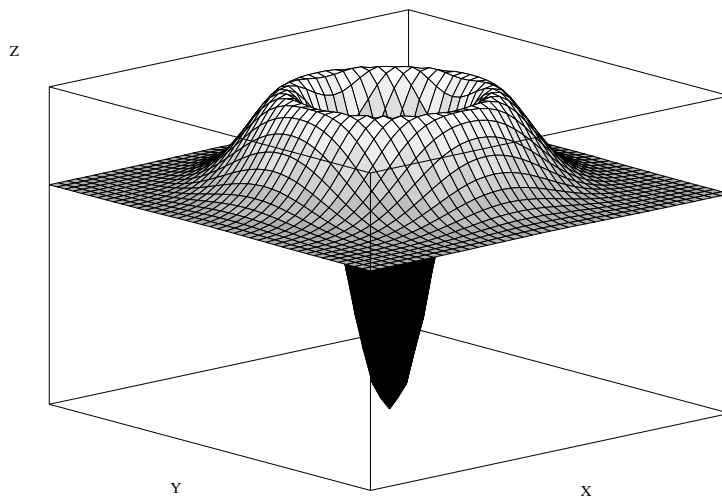


Figura 3.15: Laplaciano bidimensional de la función gaussiana ($\sigma = 1$).

La combinación del laplaciano con la función gaussiana presenta las siguientes características (Sonka et al. 1993):

- La suavización con la función gaussiana elimina la influencia de los puntos situados a una distancia inferior a 3σ del punto actual.
- El operador de Laplace $\nabla^2 G$ es una medida eficiente y estable de los cambios de la imagen.
- Los bordes son localizados en los puntos de la imagen donde la transformación indicada en 3.23 produce cruces por cero.

Algunos experimentos neurofisiológicos han evidenciado que la retina del ojo humano realiza operaciones muy similares a las operaciones realizadas por el laplaciano de la función gaussiana (Marr 1982; Ullman 1981).

El operador $\nabla^2 G$ puede ser calculado de forma aproximada utilizando máscaras de convolución correspondientes a la diferencia de dos máscaras del operador gaussiano que tengan valores de σ sustancialmente diferentes. Este método es denominado “la diferencia de gaussianos (DoG)” (Marr y Hildreth 1980; Marr 1982).

3.3.4.3 Algoritmos basados en criterios de optimización

El realce óptimo trata de diseñar detectores de bordes con un rendimiento deseado. Hay dos problemas relacionados con la detección óptima de bordes:

- La definición de los criterios de optimización.
- El diseño de un detector que optimice estos criterios.

Un buen detector de bordes ha de tener en cuenta los siguientes criterios (Canny 1986; Faugeras 1993; Sonka et al. 1993):

- Buena detección: un buen detector de bordes debe encontrar todos los bordes que haya en la imagen pero sin incluir ningún punto espurio. Debe, por tanto, ser robusto ante la presencia de ruido.
- Buena localización: la distancia entre los bordes detectados y los reales deber ser tan pequeña como sea posible.
- Unicidad de la respuesta: el detector de bordes no debe identificar múltiples bordes donde sólo hay uno.

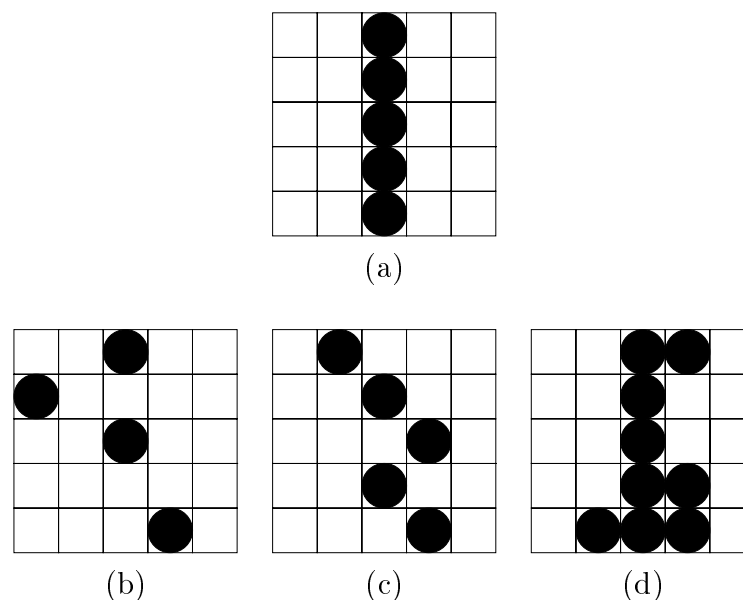


Figura 3.16: Defectos en la detección de bordes: (a) borde real, (b) poca robustez frente al ruido, (c) pobre localización y (d) múltiples respuestas.

Canny (1.986) propuso estos tres criterios para diseñar un detector de bordes óptimo para bordes de tipo escalón o salto y con la presencia de ruido blanco. El proceso realizado fue el siguiente:

- Aplicación de los dos primeros criterios a una función unidimensional para obtener una solución óptima usando cálculo de variaciones.
- Inclusión del tercer criterio (unicidad de la respuesta) que permite obtener la solución mediante optimización numérica.
- Generalización del detector a dos dimensiones.

Sea $f(x)$ una función unidimensional que represente un borde de tipo escalón:

$$f(x) = Au(x) + n(x) \quad (3.26)$$

donde

- A es la amplitud del salto del borde.
- La función $u(x)$ es una función de salto

$$u(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (3.27)$$

- La función $n(x)$ representa el ruido blanco de media 0 y varianza σ_0^2 .

Sea $h(x)$ una función desconocida que realce el impulso, $f(x)$ la función definida en 3.26 y $g(x)$ la función obtenida al convolucionar h con f :

$$g(x) = \int_{-\infty}^{+\infty} f(x-y)h(y)dy = A \int_{-\infty}^x h(y)dy + \int_{-\infty}^{+\infty} n(x-y)h(y)dy \quad (3.28)$$

En $x = 0$, se tiene

$$g(0) = A \int_{-\infty}^0 h(y)dy + \int_{-\infty}^{+\infty} n(-y)h(y)dy \quad (3.29)$$

donde el primer sumando representa la contribución de la señal (S , *signal*) y el segundo, la del ruido (N , *noise*).

La formulación matemática de los criterios utilizados para diseñar un buen detector de borde es (Canny 1986; Faugeras 1993):

- Buena detección: la probabilidad de fallo en la detección de un punto de borde real debería ser baja y también debería ser baja la probabilidad de marcar como puntos del borde a aquéllos que no lo sean. Este criterio corresponde a la maximización de la relación entre la señal y el ruido (*signal to noise ratio*, SNR).

$$SNR = \frac{S}{\sqrt{E(N^2)}} = \frac{A|\int_{-\infty}^0 h(y)dy|}{\sigma_0\sqrt{\int_{-\infty}^{+\infty} h^2(y)dy}} = \frac{A}{\sigma_0}\Sigma(h) \quad (3.30)$$

Encontrando la función $h(x)$ que maximice Σ sólo se obtiene el mejor operador para la detección de bordes.

- Buena localización: los puntos marcados por el operador como puntos del borde deberían estar tan cerca como sea posible de los puntos reales del borde. Este criterio corresponde a la minimización de la varianza σ^2 de la posición de los cruces por cero o la maximización del criterio de localización L definido como el recíproco de σ :

$$L = \frac{1}{\sqrt{E(x_0^2)}} = \frac{A|h'(0)|}{\sigma_o \sqrt{\int_{-\infty}^{+\infty} h'^2(y) dy}} = \frac{A}{\sigma_0} \Lambda(h) \quad (3.31)$$

Encontrando la función $h(x)$ que maximice Λ sólo se obtiene el mejor operador para la localización de bordes

- Unicidad de respuesta: el detector no debería producir múltiples salidas como respuesta a un único borde. Es necesario limitar el número de picos que se producen, de forma que haya una baja probabilidad de declarar más de un borde. La distancia entre los picos obtenidos al aplicar h , denotada como x_{max} , ha de ser una fracción k de la amplitud W del operador

$$x_{max} = kW = 2\pi \sqrt{\frac{\int_{-\infty}^{+\infty} h'^2(y) dy}{\int_{-\infty}^{+\infty} h''^2(y) dy}} \quad (3.32)$$

Habiendo establecido los criterios para la detección, la localización y la limitación del número de picos, Canny formuló el problema de la siguiente forma: obtener una función antisimétrica que maximice el producto

$$\Sigma(h)\Lambda(h) \quad (3.33)$$

(que es invariante a los cambios de escala o amplitud) bajo la restricción de la tercera condición.

Expresando los criterios como una composición de funciones, Canny encontró que la solución se obtenía resolviendo la siguiente ecuación diferencial (Canny 1986):

$$2h(x) - 2\lambda_2 h''(x) + 2\lambda_4 h''''(x) + \lambda_1 = 0 \quad (3.34)$$

La solución general para $x \in [-W, 0]$ puede ser escrita como:

$$h(x) = e^{\alpha x} (a_1 \sen(\omega x) + a_2 \cos(\omega x)) + e^{-\alpha x} (a_3 \sen(\omega x) + a_4 \cos(\omega x)) - \frac{\lambda_1}{2} \quad (3.35)$$

con las condiciones de contorno

$$h(0) = h(-W) = h'(W) = 0 \quad (3.36)$$

siendo

$$h'(0) = s \quad (3.37)$$

una constante desconocida.

Puesto que $h(x)$ es antisimétrica, se puede extender a todo el intervalo $[-W, W]$ usando

$$h(x) = -h(-x) \quad (3.38)$$

Al añadir la condición que indica que x_{max} es alguna fracción k de la amplitud W del operador

$$x_{max} = kW \quad (3.39)$$

se tiene que los valores de a_1 , a_2 , a_3 y a_4 son calculados como funciones de α , ω , s y λ_1 . Como resultado de este proceso, se obtiene una parametrización de h en términos de estos cuatro parámetros.

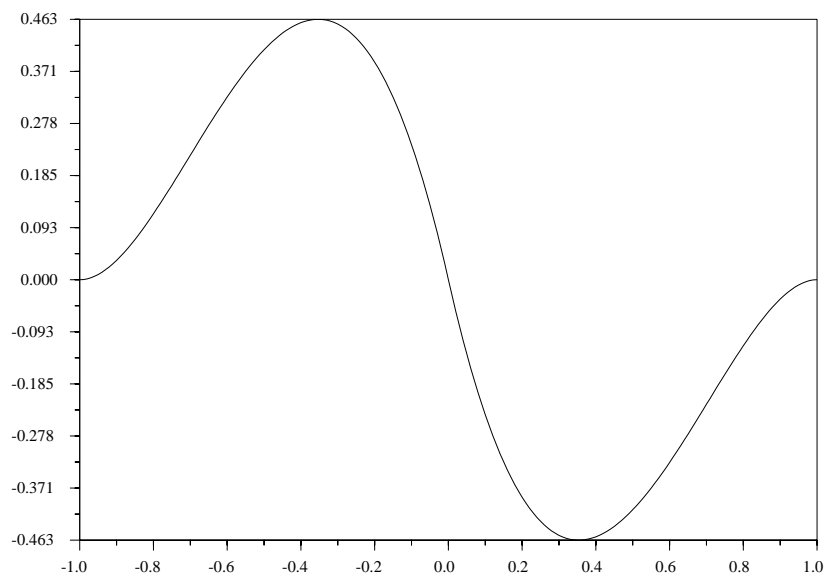
Canny encontró que el valor más grande de k que podía ser obtenido era próximo a 0.576, siendo el valor de $\Sigma(h)\Lambda(h) = 1.12$. Los valores de los parámetros que definen la función h que permite alcanzar estos valores son:

- $\alpha = 2.05220$
- $\omega = 1.516939$
- $a_1 = -1.244653939$
- $a_2 = -0.7912446531$
- $a_3 = 0.1486768717$
- $a_4 = -0.20878553476$
- $\lambda_1 = -2$

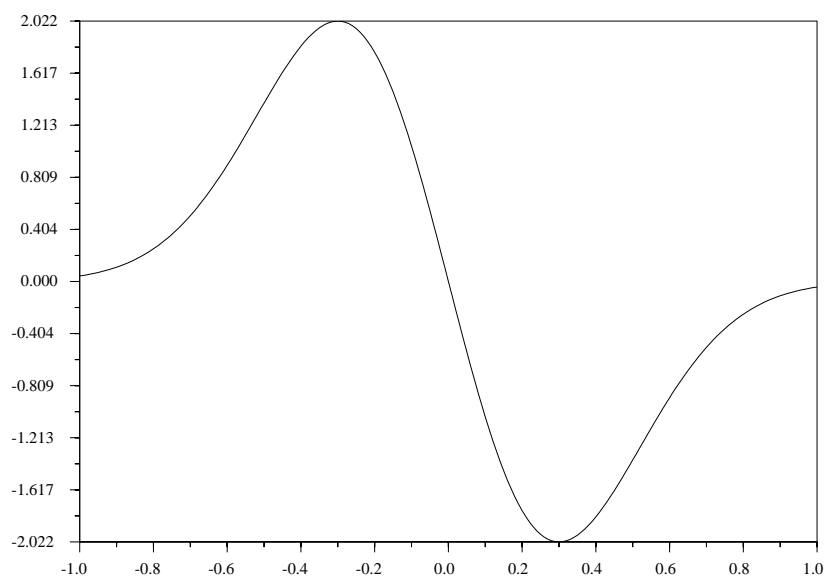
El filtro óptimo obtenido se muestra en la figura 3.17(a), donde se puede apreciar que es muy similar a la primera derivada del gaussiano (figura 3.17(b)):

$$G'(x) = \frac{-x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (3.40)$$

Si se utiliza la primera derivada de la función gaussiana, los resultados obtenidos para k y $\Sigma(h)\Lambda(h)$ son, respectivamente, 0.51 y 0.92, que no son muy diferentes de los obtenidos por el operador de Canny. Por tanto, el operador obtenido por Canny puede ser efectivamente aproximado, con un error inferior al 20%, por la primera derivada de la función gaussiana. El uso de esta aproximación está fundamentado en la existencia de una implementación efectiva del operador basado en la primera derivada de la función gaussiana (Canny 1986).



(a)



(b)

Figura 3.17: Comparación entre (a) el operador de Canny y (b) la primera derivada de la función gaussiana ($\sigma = 0.3$).

La extensión del operador de Canny a dos dimensiones consta de los siguientes pasos:

- Convolución de la imagen con un operador de una función gaussiana bidimensional simétrica.
- Posterior diferenciación en la dirección del gradiente.

Sea $G(x, y)$ una función gaussiana bidireccional simétrica

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.41)$$

y sea $G_n(x, y)$ la primera derivada de $G(x, y)$ en la dirección de \vec{n}

$$G_n(x, y) = \frac{\partial G}{\partial \vec{n}} = \vec{n} \cdot \vec{\nabla} G \quad (3.42)$$

La dirección de \vec{n} debería ser perpendicular al borde. Aunque esta dirección no es conocida a priori, se puede obtener una buena estimación de ella si la dirección suavizada del gradiente está disponible

$$\vec{n} = \frac{\vec{\nabla}(G(x, y) * f(x, y))}{|\vec{\nabla}(G(x, y) * f(x, y))|} \quad (3.43)$$

donde $f(x, y)$ representa a la imagen y el asterisco (*) denota la operación de convolución.

Un punto de borde se define como un máximo local en la dirección de \vec{n} del operador $G_n(x, y)$ aplicado a la imagen $f(x, y)$. En un máximo local, se tiene que

$$\frac{\partial}{\partial \vec{n}} G_n(x, y) * f(x, y) = 0 \quad (3.44)$$

Sustituyendo en 3.44 el valor de $G_n(x, y)$ indicado en 3.42 se obtiene

$$\frac{\partial^2}{\partial \vec{n}^2} G(x, y) * f(x, y) = 0 \quad (3.45)$$

Esta ecuación (3.45) muestra cómo encontrar el máximo local en la dirección perpendicular al borde. Esta operación se denomina “supresión de puntos que no son máximos locales” (*non-maximal suppression*).

Como la convolución y la derivada son operaciones asociativas respecto de la convolución, primero se puede convolucionar una imagen $f(x, y)$ con una función gaussiana simétrica $G(x, y)$ y después calcular la segunda derivada direccional usando una estimación de la dirección \vec{n} obtenida según la ecuación 3.43. La

fuerza del borde (la magnitud del gradiente de la imagen $f(x, y)$) es medida como

$$||(G_n(x, y) * f(x, y))|| = ||\vec{\nabla}(G(x, y) * f(x, y))|| \quad (3.46)$$

El detector óptimo de bordes propuesto por Canny ha ejercido una gran influencia sobre otros investigadores (Deriche 1987; Lee et al. 1993; Petrou y Kittler 1991; Sarkar y Boyer 1991; Shen y Castan 1986; Spacek 1986; Tagare y de Figueiredo 1990).

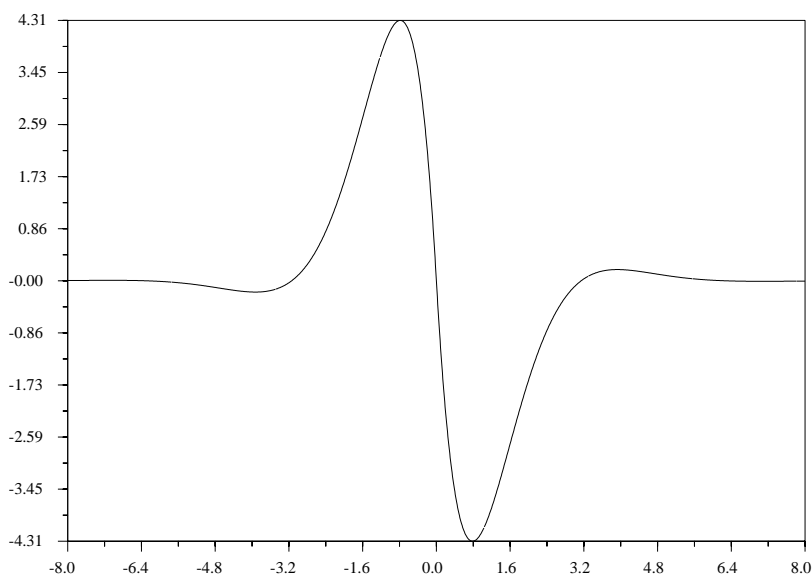


Figura 3.18: Filtro de Deriche ($\alpha = \omega = 1$ y $a_1 = -13.3816$).

Deriche (1987) extiende la función $h(x)$ a toda la recta real, eliminando la restricción que limitaba la definición de $h(x)$ al intervalo $[-W, W]$. Se obtiene, por tanto, un filtro de respuesta infinita a los impulsos (*infinite impulse response, IIR*). Asumiendo sin pérdida de generalidad que $\alpha \geq 0$, la condición $h(-\infty) = 0$ implica que $a_3 = a_4 = \lambda_1 = 0$. Las otras condiciones implican que $a_2 = 0$ y $a_1\omega = s$. La solución obtenida es (figura 3.18):

$$h(x) = \begin{cases} a_1 e^{\alpha x} \text{sen}(\omega x) & x < 0 \\ a_1 e^{-\alpha x} \text{sen}(\omega x) & x \geq 0 \end{cases} \quad (3.47)$$

o

$$h(x) = a_1 e^{-\alpha|x|} \text{sen}(\omega x) \quad (3.48)$$

El valor de a_1 debe ser negativo.

Deriche criticó que Canny sólo utilizara la forma de su operador para justificar el uso de la primera derivada de la función gaussiana como aproximación

válida. Deriche no utiliza tal aproximación, sino que hace uso de una implementación recursiva para extender su operador al caso bidimensional (Deriche 1987; Faugeras 1993).

Spacek (1986) incorporó que, además del producto indicado en la ecuación 3.33, también se debía maximizar la expresión 3.32 (Faugeras 1993; Spacek 1986). La solución que obtuvo fue (figura 3.19):

$$h(x) = e^{\alpha x}(a_1 \sin(\alpha x) + a_2 \cos(\alpha x)) + e^{-\alpha x}(a_3 \sin(\alpha x) + a_4 \cos(\alpha x)) - \frac{\lambda_1}{2} \quad (3.49)$$

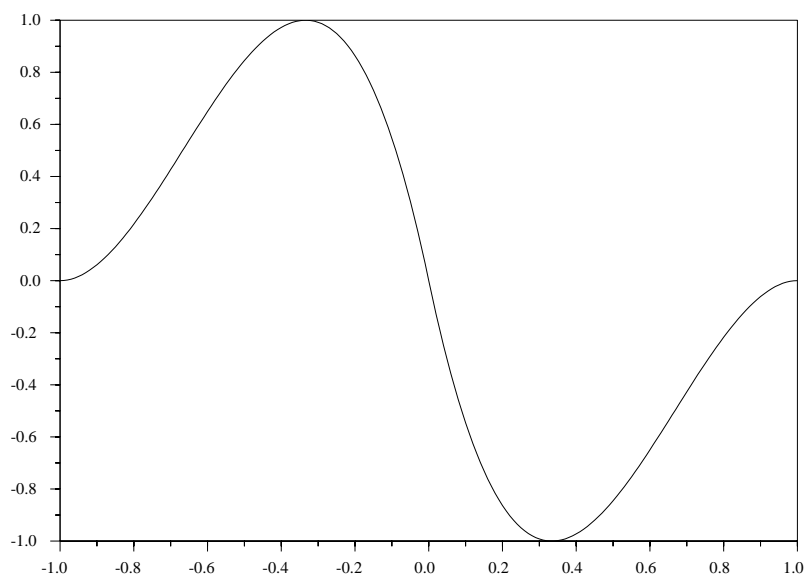


Figura 3.19: Filtro de Spacek.

Shen y Castan (1986) propusieron utilizar el filtro de suavización de la función exponencial simétrica (sección 2.5, función 2.29) para diseñar dos métodos de detección de bordes, uno basado en el máximo del gradiente y otro que usaba los cruces por cero de la segunda derivada en la dirección del gradiente. Al igual que Deriche, hicieron una implementación recursiva de sus detectores de bordes.

Shen y Castan afirmaron que sus filtros daban mejores resultados al detectar y localizar los bordes pero no indicaron nada acerca de la multiplicidad de respuestas.

Una evaluación de los filtros propuestos por Canny, Deriche y Shen y Castan puede ser encontrada en (Castan et al. 1990; Ziou y Fabre 1994; Ziou y Wang 1996).

Se han diseñado muchos detectores de bordes utilizando criterios de optimización y fundamentos teóricos diferentes (Basseville et al. 1981; Basseville 1981; Huertas y Medioni 1986; Kisworo et al. 1994; Martelli 1972; Modestino y Fries 1977; Monga et al. 1991; Montanari 1971; Petrou y Kittler 1991; Shanmugam et al. 1979; Sarkar y Boyer 1991; Sarkar y Boyer 1994; Tabbone y Ziou 1992; Tabbone 1994b; Tagare y de Figueiredo 1990; Tagare y de Figueiredo 1994).

3.3.5 Identificación de los bordes

La identificación de los bordes (*edge labeling*, etiquetaje del borde) tiene como cometidos:

- La localización de los bordes.
- La supresión de los bordes falsos.

La identificación de los bordes depende de los operadores utilizados en la suavización y la diferenciación:

- Métodos basados en el gradiente: los bordes son localizados en los puntos donde la magnitud del gradiente alcanza un máximo local.
- Métodos basados en las derivadas de segundo orden: los bordes son localizados en los puntos donde se produce un cruce por cero ("zero crossing") de la derivada de segundo orden.

La eliminación de los bordes falsos incrementa la proporción de la señal frente al ruido (*signal to noise ratio*) en la imagen obtenida con las operaciones de suavización y diferenciación. A pesar de la importancia de esta operación de limpieza, se han hecho pocos trabajos específicos sobre esta materia (Ziou y Tabbone 1998).

3.3.5.1 Identificación de bordes en métodos basados en el gradiente

Al aplicar un método de detección de bordes basado en el gradiente, los puntos de la imagen que pertenecen a un borde tienen una respuesta mucho mayor que la de aquellos puntos que pertenecen a zonas de intensidad uniforme, como el fondo de la imagen o el interior de los objetos. Sin embargo, este realce no es suficiente, siendo necesaria la obtención de una imagen binaria para que el proceso de segmentación se dé por terminado. La imagen de salida ha de ser una imagen binaria, es decir, compuesta por puntos que sólo pueden tener un nivel de gris entre dos posibles, uno para representar los bordes (normalmente en blanco)

y otro para el resto de la imagen (en negro). El método clásico para generar esta imagen binaria ha sido la umbralización del histograma de niveles de gris correspondiente al módulo del gradiente de los puntos de la imagen.

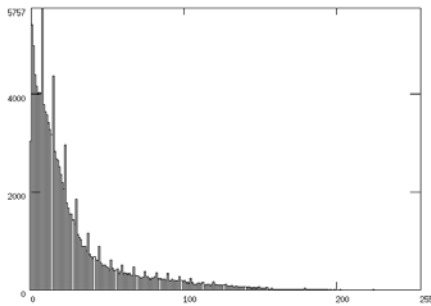
La umbralización de la imagen debe seleccionar un determinado nivel de intensidad, denominado “umbral”, que sirva como límite de separación entre lo que se considera como borde y lo que no, de forma que los puntos de la imagen con un nivel de intensidad inferior a dicho umbral se considerarán parte del fondo de la imagen o del interior de los objetos, mientras que los puntos con un nivel de intensidad superior formarán parte de los bordes.



(a)



(b)



(c)



(d)

Figura 3.20: (a) Imagen monocromática, (b) imagen obtenida después de la suavización y diferenciación, (c) histograma de (b), y (d) umbralización con $U = 50$.

Si $g(x, y)$ es la imagen obtenida al aplicar un detector de bordes a una imagen $f(x, y)$, una operación de umbralización de bordes produce una imagen binaria $b(x, y)$ de la siguiente forma:

$$b(x, y) = \begin{cases} 0 & \text{si } g(x, y) < U \\ 1 & \text{si } g(x, y) \geq U \end{cases} \quad (3.50)$$

donde U representa el umbral seleccionado, 0, el color negro, y 1, el color blanco.

La estrategia utilizada en la determinación de los umbrales permite clasificar los métodos de umbralización en tres tipos (Castleman 1979; González y Woods 1996; Pal y Pal 1993):

- Global: se selecciona un único umbral que sólo depende de $g(x, y)$ y es válido para toda la imagen.
- Local o adaptativa: los umbrales dependen de $g(x, y)$ y de propiedades locales de los puntos, como, por ejemplo, el valor medio de intensidad de un vecindario centrado en (x, y) .
- Dinámica: si, además de lo anterior, los umbrales dependen de las coordenadas (x, y) de cada punto.

La mayoría de los métodos de detección de bordes han utilizado un umbral global pero también se han usado umbrales locales (Haralick y Shanmugan 1973; Haralick y Shapiro 1985; Levine 1985; Pitas 1993).

Son muchos y muy variados los fundamentos teóricos que se han utilizado para seleccionar correctamente los umbrales de un histograma. A continuación se citan algunos de ellos:

- El porcentaje de puntos de los bordes de los objetos respecto del total de la imagen (Pratt 1978; Sonka et al. 1993).
- La media y la varianza del ruido (Haddon 1987).
- La búsqueda de mínimos (Maravall 1993).
- Las técnicas de reconocimiento de formas (Maravall 1993).
- La entropía (Abutaleb 1989; Brink 1995; Kapur et al. 1985; Li y Lee 1993; Liscano y Wong 199x).
- La concavidad o convexidad del histograma (Rosenfeld y Torre 1983; Whatmough 1991).
- La agrupación de clases o *clustering* del histograma (Kittler y Illingworth 1985; Spann y Horne 1989; Tsai y Chen 1992).
- Umbralización óptima o umbralización basada en la distribución de la función de densidad de probabilidad (González y Woods 1996; Sonka et al. 1993).
- Teoría de conjuntos borrosos (Chi y Yan 1993).
- Conjuntos aleatorios (Friel y Molchanov 1999).

- Redes Neuronales (Peak y Tag 1994).

Sin embargo, el método de umbralización que, desde su aparición, ha sido más ampliamente utilizado en la detección de bordes es el algoritmo de histéresis (Canny 1986). Este algoritmo usa dos umbrales, uno inferior y otro superior, de forma que un borde compuesto por una lista de puntos conectados es considerado como verdadero si el valor de cada punto de la lista está por encima del umbral inferior y al menos uno está por encima de un umbral superior; en caso contrario, el borde es falso.

Se han realizado importantes recopilaciones (Pal y Pal 1993; Sahoo et al. 1985; Weszka 1978) y estudios comparativos (Glasbey 1993; Lee y Chung 1990; Leung y Lam 1996) sobre los algoritmos de umbralización.

La principal dificultad de la umbralización consiste en la correcta selección del umbral. Generalmente, el valor del umbral ha sido encontrado usando un proceso de “ensayo y error”. Sin embargo, el umbral depende de las características del tipo de borde, de las propiedades del filtro de suavización y de las propiedades del operador de diferenciación (Dellepiane et al. 1989; Tabbone y Ziou 1993; Ziou y Tabbone 1993; Ziou y Tabbone 1995).

La umbralización es una técnica sencilla y con un coste computacional muy bajo, pero tiene dos inconvenientes cuando se utiliza para detectar los bordes de una imagen:

- Si se elige un umbral muy bajo, los bordes localizados suelen tener un grosor de más de un punto, siendo necesaria una fase posterior de adelgazamiento³ (*thinning*) o esqueletización (*skeletization*) (figura 3.21).
- Si se elige un umbral muy alto, los bordes localizados pueden estar desconectados o “rotos”, requiriéndose, por tanto, un enlazamiento de bordes (*edge linking*).

Se puede evitar el adelgazamiento de los bordes si antes de umbralizar la imagen se ejecuta el algoritmo de supresión de puntos que no son máximos locales. La idea básica consiste en extraer los máximos locales del módulo del gradiente. Un algoritmo eficiente fue propuesto por Canny (1983) y consiste en encontrar el máximo local a lo largo de la dirección del vector gradiente. Un punto es un máximo local si el módulo del gradiente en ese punto es más grande que el módulo del gradiente en dos puntos vecinos situados a la misma distancia a uno y otro lado del punto considerado en la dirección del gradiente (Faugeras 1993; Niblack 1986). Para realizar esta comprobación, se han de realizar dos acciones:

- Decidir el tamaño del vecindario del punto en el cual se desea comprobar si es un máximo local.

³Véase el capítulo 5.



Figura 3.21: (a) Imagen generada por un detector de bordes e (b) imagen obtenida al realizar el adelgazamiento de (a) con el algoritmo de Wang-Zhang (1.989) modificado (Krishnapuram y Chen 1993; Wang y Zhang 1989).

- Interpolan los valores del módulo del gradiente en aquellos puntos del vecindario en los que corte la recta que pasa por el punto central y que tiene como dirección su vector gradiente. Esta interpolación es necesaria puesto que los puntos obtenidos en la intersección pueden no ser reales.

Generalmente, el tamaño del vecindario elegido es de 3×3 y los valores del módulo del gradiente son interpolados linealmente.

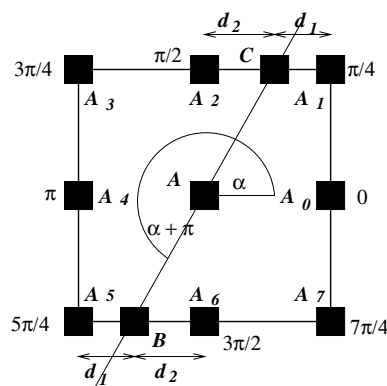


Figura 3.22: Vecindario de tamaño 3×3 utilizado para comprobar si el punto central es un máximo local.

Obsérvese la figura 3.22, donde el punto central tiene como magnitud y dirección del gradiente a A y α , respectivamente. Las magnitudes del gradiente de los puntos del vecindario son $A_0, A_1, A_2, A_3, A_4, A_5, A_6$ y A_7 . Para verificar si el punto central es un máximo local, se debe comparar A con los valores B y C , que son las magnitudes del gradiente de los puntos obtenidos al intersectar el vecindario con la recta que tiene dirección α y pasa por el punto central. El valor de B es calculado mediante interpolación lineal de los valores A_5 y A_6 . Una

interpolación análoga es realizada con los valores A_1 y A_2 para calcular el valor de C . El punto central es considerado como un punto de borde si

$$A > B \wedge A > C \quad (3.51)$$

Si no se cumple esta condición, se asigna a A un valor igual a cero; en caso contrario, existen dos posibilidades: mantener el valor de A y realizar posteriormente una umbralización (Sonka et al. 1993) o asignarle directamente a A el valor máximo que se hubiera utilizado en el proceso de umbralización (Faugeras 1993), considerando como puntos de los bordes a todos los máximos locales de la imagen. En ambos casos, el resultado final es una imagen binaria donde los puntos marcados corresponden a los bordes. La figura 3.23(c) muestra el resultado obtenido al utilizar la primera variante. Se puede apreciar que el grosor de los bordes obtenidos es sólo de un punto.

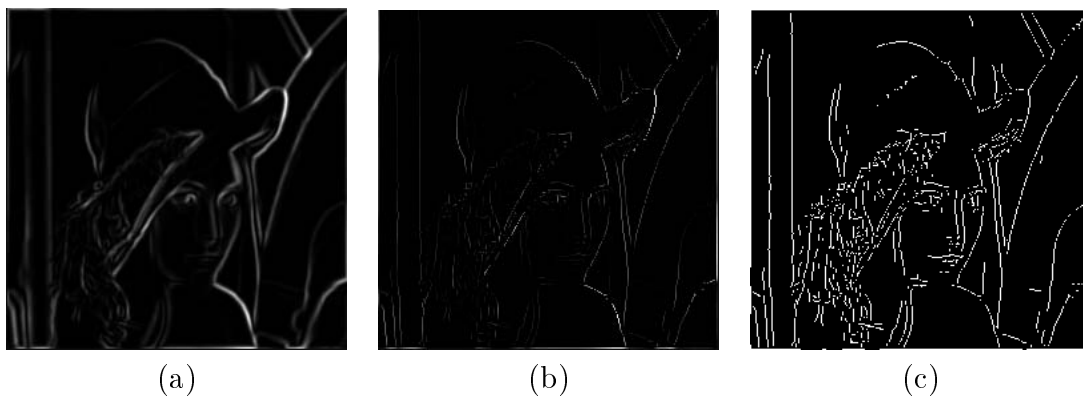


Figura 3.23: (a) Gradiente de una imagen, (b) gradiente de la misma imagen habiendo suprimido los puntos que no son máximos locales y (c) umbralización de (b).

El segundo inconveniente, relativo a los bordes desconectados o “rotos”, hace necesaria una fase de enlazamiento de bordes (González y Woods 1996; Pitas 1993). La mayoría de los métodos de enlazamiento diseñados han hecho uso de la información suministrada por el operador de diferenciación para estimar el gradiente, el cual indica la dirección perpendicular a la dirección en la que debe producirse el enlazamiento (Atae-allah et al. 1997; Canny 1986; Farag y Delp 1995; Rosenfeld y Kak 1982a; Wang 1994). Sin embargo, también es posible diseñar algoritmos de enlazamiento de bordes que sólo utilicen la distribución geométrica de los puntos de la imagen (Xie 1992). Hough (1.962) propuso una técnica global para realizar el enlazamiento, denominada “la transformada de Hough” (*Hough transform*), que consideraba las relaciones globales entre los puntos. El método propuesto por Hough fue popularizado por Duda y Hart (1.972).

3.3.5.2 Identificación de bordes en métodos basados en las derivadas de segundo orden

Estos métodos localizan los bordes en los puntos donde se producen los cruces por cero (*zero crossings*) o cambios de signo de las derivadas de segundo orden. El proceso consiste en comparar un punto con los puntos vecinos que estén a la izquierda y debajo de él. Si estos tres puntos no tienen el mismo signo, entonces hay un cruce por cero. Se pueden utilizar múltiples direcciones para mejorar la localización, especialmente la de ciertos modelos de uniones (*junctions*) (v.g.: L-uni3n) (Tabbone 1994b). Huertas y Medioni (1.986) propusieron el uso de once posibles predicados para comprobar la existencia de un cruce por cero.

La detecci3n de bordes mediante derivadas de segundo orden presenta como principal inconveniente la localizaci3n de bordes falsos, es decir, de falsos cruces por cero. Algunos autores han utilizado el m3dulo del gradiente para establecer una medida de admisibilidad (Clark 1989; Shen y Castan 1986). Sin embargo, es bien conocido que un aut3ntico cruce por cero puede corresponder a “un punto de silla” del m3dulo del gradiente. Otros autores (Hildreth 1983; Marr y Hildreth 1980; Tabbone y Ziou 1991; Tabbone 1994b; Ziou y Tabbone 1993) sugieren el uso de la pendiente del borde como medida de admisibilidad. Sin embargo, la computaci3n de la pendiente requiere derivadas de tercer orden y, por tanto, son sensibles al ruido.

Los bordes falsos asociados a falsos cruces por cero pueden ser de dos clases: (Clark 1988; Clark 1989; Tabbone y Ziou 1991; Tabbone 1994b; Ziou y Tabbone 1993):

- Bordes “ruidosos”.
- Bordes “fantasmas”.

La primera clase de bordes falsos es muy conocida y su origen est3 en el ruido povocado durante la adquisici3n de la imagen o por la naturaleza de la escena, es decir, la textura. En muchas im3genes, este tipo de bordes falsos tiene generalmente un m3dulo del gradiente bajo y pueden ser descartados usando un algoritmo de umbralizaci3n como el de hist3resis.

Los bordes “fantasmas” surgen a partir de ciertos modelos de bordes como los de tipo “escalera”. Intuitivamente, son cruces por cero de la segunda derivada que corresponden al m3nimo positivo o al m3ximo negativo de la primera derivada del borde de tipo escalera.

Se indica en (Clark 1989) que, para el caso de se3ales aleatorias normalmente distribuidas, los bordes fantasmas son m3s peque3os y m3s d3biles que los aut3nticos bordes. En la pr3ctica, sin embargo, puede haber bordes fantasmas cuya magnitud del gradiente sea mayor que algunos de los aut3nticos bordes.

Los bordes fantasmas son los únicos mínimos de la magnitud del gradiente que verifican las siguientes condiciones:

$$\frac{\partial^2(f * g)'_{\vec{n}}(x_0, y_0)}{\partial x^2} \frac{\partial^2(f * g)'_{\vec{n}}(x_0, y_0)}{\partial y^2} - \left(\frac{\partial^2(f * g)'_{\vec{n}}(x_0, y_0)}{\partial x \partial y} \right)^2 > 0 \quad (3.52)$$

$$\frac{\partial^2(f * g)'_{\vec{n}}(x_0, y_0)}{\partial x^2} > 0 \quad (3.53)$$

donde $f(x, y)$ es la imagen de entrada, $g(x, y)$ es el filtro de suavización utilizado, $(f * g)'_{\vec{n}}(x, y)$ es la primera derivada de $(f * g)(x, y)$ tomada en la dirección del gradiente \vec{n} .

En resumen, el algoritmo de eliminación de los bordes falsos consta de dos fases:

- Eliminación de los bordes ruidosos usando el algoritmo de histéresis o cualquier otro algoritmo de umbralización.
- Eliminación de los bordes fantasmas utilizando las condiciones 3.52 y 3.53

Se ha propuesto una regla automática para calcular el umbral del detector laplaciano de la función gaussiana (Dellepiane et al. 1989), pero esta regla es empírica, no se ha dado ninguna justificación de su uso y sólo ha sido probada sobre imágenes sintéticas (Ziou y Tabbone 1998).

También se ha propuesto un algoritmo de limpieza de bordes falsos en algoritmos de detección de bordes multiescala, que está basado en el comportamiento de un borde ideal de tipo escalón (Tabbone y Ziou 1991; Tabbone y Ziou 1993; Ziou y Tabbone 1993). El umbral es encontrado en una escala alta y es propagado automáticamente a los bordes ideales de tipo escalón obtenidos en las escalas más bajas. Mejoras de este algoritmo se han hecho en (Ziou y Tabbone 1995) para que se pueda usar con cualquier filtro de suavización, operador de diferenciación y modelo de borde.

3.3.6 Multidetectores y multiescalas

Un método de detección de bordes debería especificar (Ziou y Tabbone 1998):

- Las características de las imágenes que permiten un funcionamiento más eficiente del detector de bordes.
- La escala de computación que ha de utilizarse. Muchos detectores pueden ser descritos como una operación de convolución

$$(g_s * f)(x, y) \quad (3.54)$$

donde f es la imagen, g_s es el filtro y s es la escala.

La mayoría de los métodos de detección de bordes no hacen referencia a ninguno de estos aspectos, siendo necesario, por tanto, la selección del método más adecuado para cada aplicación. Generalmente, se suele elegir arbitrariamente un detector, independientemente de las características de la imagen, el cual es utilizado para encontrar todos los bordes de la imagen que esté siendo procesada. El problema de la selección de la escala es resuelto mediante experimentos de ensayo y error, utilizándose el valor obtenido para todas las imágenes. Es obvio que los resultados obtenidos distan mucho de ser correctos. De hecho, un detector que se ejecute en una escala no produce todos los bordes de la imagen (Ziou y Koukam 1995).

Una solución más adecuada consiste en utilizar varios detectores de bordes que difieran en sus escalas, propiedades matemáticas y objetivos. De hecho, ya existen algunos sistemas de procesamiento de imágenes que usan múltiples detectores y múltiples escalas para detectar los bordes de las imágenes (Hasegawa et al. 1996; Ziou y Koukam 1995).

Una aproximación a esta solución es la detección de bordes con mutiescala, que consiste en utilizar un único detector de bordes con múltiples escalas (Ziou y Tabbone 1998). Considérese el gradiente del gaussiano:

- Si se utiliza una escala pequeña, se obtienen los bordes con más precisión, pero la influencia del ruido es mucho mayor.
- Si se utiliza una escala grande, se pueden obtener los bordes correspondientes a los cambios bruscos de intensidad, pero aparecen errores de localización.

Es difícil encontrar una única escala de un detector que sea óptima para todos los bordes de una imagen. No obstante, se puede obtener una buena descripción de los bordes de la imagen si se combinan los resultados obtenidos al aplicar el detector de bordes con diferentes escalas (Marr y Hildreth 1980; Rosenfeld y Thurston 1971). Esta técnica se denomina “aproximación multiescala” y ha sido utilizada por muchos métodos de detección de bordes (Bergholm 1987; Lu y Jain 1989; Lu y Jain 1992; Perona y Malik 1990; Williams y Shah 1990; Ziou y Tabbone 1993).

El uso de los detectores multiescala plantea los siguientes problemas:

- Determinar el número de escalas que se van a utilizar.
- Seleccionar el orden en que las escalas serán aplicadas.
- Combinar de manera eficiente la información obtenida en las diferentes escalas.

Los métodos de detección multiescala pueden utilizar dos estrategias de ejecución:

- Descendente (*coarse to fine strategy*): el detector de bordes comienza a aplicarse con escalas grandes, para, a continuación, ir reduciendo las escalas poco a poco. Esto implica que primero se extraen los bordes más relevantes y después los más sutiles (Canny 1983; Canny 1986; Lacroix 1990; Lindeberg 1993; Lindeberg 1994; Lindeberg 1996).
- Ascendente (*fine to coarse strategy*): se realiza el proceso inverso (Bergholm 1987; Williams y Shah 1990).

Tabbone y Ziou han diseñado un algoritmo que no utiliza ninguna de estas estrategias para combinar la información de los bordes: se consideran cuatro tipos de bordes (ideal, borroso, pulso, escalera) y se utilizan dos escalas (alta y baja) para conseguir una descripción completa y correcta de la imagen en términos de los cruces por cero (Tabbone y Ziou 1993; Ziou y Tabbone 1993).

Por otro lado, hay ciertos bordes que son invariantes al espacio de escala y no necesitan ser procesados en cada escala. Este problema de redundancia puede ser evitado si es posible predecir las escalas que provocan que un borde cambie. Se han propuesto varios métodos de estimación automática de escalas (Elder y Zucker 1996; Elder y Zucker 1998; Gokmen y Li 1993; Jeong y Kim 1992; Korn 1988; Lunscher y Beddoes 1986; Lindeberg 1994; Lindeberg 1996).

3.4 Detección de bordes en imágenes en color

3.4.1 Características de los algoritmos de detección de bordes en imágenes en color

Los algoritmos de detección de bordes en color pueden ser caracterizados por:

- El espacio de color⁴ que utilizan: la mayoría de los detectores de bordes pueden ser utilizados sobre cualquier espacio de color, pero algunos han sido diseñados para ser aplicados a imágenes definidas en algún espacio de color concreto.
- El filtro de suavización que aplican para atenuar el ruido: todos los detectores de bordes pueden ser complementados con una fase previa de suavización que permita atenuar el ruido a la vez que se preservan los bordes.

⁴El apéndice A describe las características de los espacios de color.

- Los fundamentos teóricos en los que se basan: se han utilizado diferentes técnicas para detectar los bordes en imágenes en color. Algunas de estas técnicas son extensiones o variantes de técnicas monocromáticas, mientras que otras han sido diseñadas específicamente para imágenes en color o multispectrales.

3.4.1.1 Espacios de color utilizados por los algoritmos de detección de bordes

Muchos detectores de bordes en color han sido diseñados sin tener en cuenta el espacio de color en el que pudieran estar definidas las imágenes. Estos detectores de bordes en color pueden ser aplicados a cualquier imagen definida sobre cualquier espacio de color. No obstante, sus autores han utilizado algún espacio de color concreto para comprobar el rendimiento del detector diseñado.

La inmensa mayoría han utilizado el espacio de color *RGB* (Alshatti y Lambert 1993; Chapron 1992; Chapron 1997; Cumani et al. 1991; Fotinos et al. 1999; Macaire et al. 1996; Novak y Shafer 1987; Pietikäinen y Harwood 1986; Robinson 1976; Robinson 1977a; Scharcanski y Venetsanopoulos 1997; Tao y Huang 1997; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Wright 1989; Zenzo 1986), pero también se han utilizado otros espacios de color como YT_1T_2 (Nevatia 1977), *rgb* (Yachida y Tsuji 1970; Skarbek y Koschan 1994) o el espacio de los colores contrarios (*opponent colors space*) ($R - G, Y - B$) (Young 1986a).

Incluso algunos autores han utilizado varios espacios de color para contrastar el rendimiento de su detector de bordes (Daily 1989; Gauch y Hsia 1992; Huntsberger y Descalzi 1985; Wesolkowski y Jernigan 1999).

Otros detectores de bordes en color han sido diseñados para imágenes definidas en algún espacio de color concreto, intentando con ello aprovechar las características específicas de dicho espacio de color:

- Se han utilizado las componentes *I* y *Q* del espacio de color *YIQ* para detectar los bordes provocados sólo por cambios de color (Machuca y Philips 1983).
- Se ha propuesto un método, basado en el espacio de color *HSV*, que evita detectar los bordes provocados por las reflexiones especulares (Tsang y Tsang 1996; Tsang y Tsang 1997).
- Se ha tenido en cuenta la correlación entre las componentes del espacio de color *RGB* para diseñar detectores de bordes (Krishnamoorthi y Bhattacharyya 1997; Li y Wu 1997).

- Para evitar la correlación de las componentes del espacio de color *RGB*, se han diseñado algoritmos que utilizan espacios de color perceptualmente uniformes, los cuales separan la información cromática de los niveles de iluminación o intensidad, como los espacios de color *HSI* (Carron y Lambert 1994; Fuertes et al. 1997; Pérez y Koch 1994) o *Lab* (Lanser 1993; Ruzon y Tomasi 1999).
- Se ha diseñado un detector agregado que utiliza de forma conjunta los espacios de color *HSI* y *RGB* (Moghaddamzadeh y Bourbakis 1997).

Estos detectores de bordes también pueden ser aplicados a imágenes definidas en otros espacios de color, pero puede que su rendimiento no sea el deseado.

3.4.1.2 Suavización de imágenes en color

Las imágenes en color, al igual que las imágenes monocromáticas, pueden contener diferentes tipos de ruido, lo cual dificulta la detección de los bordes. Para suprimir o atenuar el ruido en una imagen en color, se pueden utilizar los métodos de suavización descritos en la sección 2.5, y especialmente los métodos basados en los filtros vectoriales direccionales (Plataniotis et al. 1997; Trahanias et al. 1996).

3.4.1.3 Fundamentos teóricos de los detectores de bordes en color

Atendiendo a los fundamentos teóricos utilizados, las técnicas de detección de bordes en imágenes en color pueden ser clasificadas como:

- Detectores basados en técnicas monocromáticas: los detectores monocromáticos de bordes son utilizados o adaptados para ser aplicados a imágenes en color.
- Detectores que consideran a las imágenes en color como campos vectoriales: una imagen en color es definida como una función (un campo vectorial) que asocia a cada punto (x, y) un vector tridimensional del espacio de color utilizado.

A continuación se van a describir las características fundamentales de cada uno de estos tipos de detección de bordes en color, indicando expresamente el método de identificación de bordes utilizado en cada caso.

3.4.2 Detectores de bordes en color basados en técnicas monocromáticas

Variantes o extensiones de las técnicas monocromáticas de detección de bordes han sido utilizadas para diseñar diferentes detectores de bordes en imágenes en color. Estos detectores se pueden agrupar en cuatro clases:

- Detectores que no utilizan la información cromática de la imagen en color.
- Detectores basados en la reducción de la dimensionalidad de la imagen en color.
- Detectores basados en la fusión de bordes.
- Detectores basados en los gradientes monocromáticos.

3.4.2.1 Detectores que no utilizan la información cromática de la imagen en color

La estrategia más simple para detectar bordes en una imagen en color consiste en utilizar sólo la información acromática suministrada por los niveles de intensidad y aplicar, a continuación, cualquiera de las técnicas monocromáticas de detección de bordes descritas en la sección 3.3. Obviamente, esta estrategia no representa ninguna mejora de las técnicas monocromáticas, ya que sólo permite detectar bordes debidos a discontinuidades en los niveles de intensidad.

Dependiendo del tipo de espacio de color utilizado, la información acromática de la imagen en color puede estar directamente disponible en alguna de sus componentes, como ocurre con la componente I del espacio de color *HSI* o la componente Y del espacio de color *YIQ*, o ha de obtenerse mediante alguna combinación lineal de las componentes del espacio de color, como ocurre con las imágenes del espacio de color *RGB*, donde la imagen acromática suele ser definida como la media aritmética de sus componentes de color (Koschan 1995; Ohta et al. 1980; Ohta 1985):

$$I(x, y) = \frac{1}{3}(R(x, y) + G(x, y) + B(x, y)) \quad (3.55)$$

Uno de los algoritmos que utiliza esta estrategia está basado en un operador borroso (Chen et al. 1995).

3.4.2.2 Detectores basados en la reducción de la dimensionalidad de la imagen en color

Algunos detectores reducen la información tridimensional de la imagen en color a una única información unidimensional, para, a continuación, aplicar un detector monocromático de bordes. Esta estrategia reduce el problema de la detección de bordes en color a un problema de detección monocromática. Por esta razón, es muy cuestionable que esta estrategia suponga alguna mejora respecto de los métodos monocromáticos (Luong 1991; Nevatia 1977).

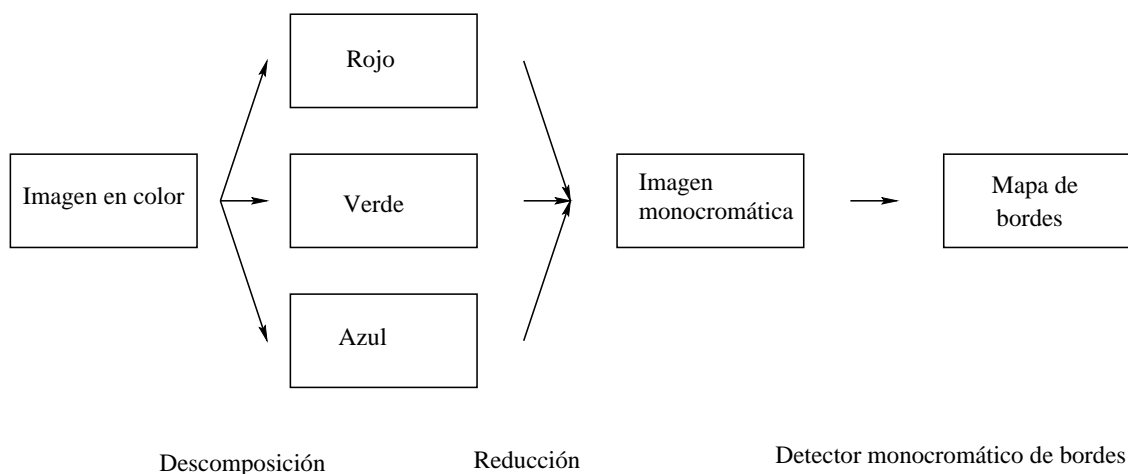


Figura 3.24: Fases de la detección de bordes en color mediante reducción de la dimensionalidad.

Algunos de los métodos utilizados para reducir la información tridimensional están basados en:

- Métrica de color: algunos detectores hacen uso de alguna métrica en el espacio de color para definir la noción de distancia colorimétrica. Los bordes son detectados en los puntos donde se produce alguna discontinuidad de la distancia definida (Claxton y Kwok 1987; Macaire et al. 1996; Nevatia 1977; Novak y Shafer 1987).
- Proyección sobre un eje: la información de color de cada punto es considerada como un vector tridimensional que puede ser proyectado sobre un eje. Para definir el eje de proyección se han propuesto diferentes métodos (Yang y Tsai 1996):
 - Expansión de Karhunen - Loeve (Bigun 1993; Fukunaga 1990): en cada bloque o entorno de $n \times n$ puntos de la imagen en color se calcula

la siguiente matriz de covarianzas:

$$\Sigma = \begin{bmatrix} E(R^2) - E^2(R) & E(RG) - E(R)E(G) & E(RB) - E(R)E(B) \\ E(RG) - E(R)E(G) & E(G^2) - E^2(G) & E(GB) - E(G)E(B) \\ E(RB) - E(R)E(B) & E(GB) - E(G)E(B) & E(B^2) - E^2(B) \end{bmatrix} \quad (3.56)$$

Σ tiene autovalores no negativos puesto que es una matriz de covarianza. Se elige como eje proyección el autovector asociado al autovalor mayor.

- Utilizar como eje de proyección el vector mediana de cada bloque de puntos (Astola et al. 1990).
- Aplicar en cada bloque de puntos el algoritmo de umbralización de dos niveles que preserva los momentos de la imagen (Tai 1985) para obtener dos vectores

$$\vec{v}_1 = (R_1, G_1, B_1) \quad (3.57)$$

y

$$\vec{v}_2 = (R_2, G_2, B_2) \quad (3.58)$$

El eje de proyección es definido por el vector

$$\vec{v} = \vec{v}_1 - \vec{v}_2 \quad (3.59)$$

Yang y Tsai (1.996) utilizaron este último método junto con un detector monocromático de bordes, denominado “*mass - moment edge detector*” (Tabatai y Mitchell 1984), para detectar bordes en una imagen en color.

3.4.2.3 Detectores basados en la fusión de bordes

Esta estrategia de detección de bordes consta de las siguientes fases (Koschan 1995; Plataniotis y Venetsanopoulos 2000; Ruzon y Tomasi 1999; Zhu et al. 1999):

1. Descomposición de la imagen en sus componentes de color.
2. Aplicación de una técnica monocromática de detección a cada una de las componentes de la imagen en color de forma independiente.
3. Fusión de los bordes detectados en cada una de las componentes de color utilizando alguna técnica apropiada.

Debido a esta fusión final de los resultados, los métodos basados en esta técnica también se denominan “métodos de fusión de salidas” (*outputs fusion methods*) (Ruzon y Tomasi 1999). La figura 3.25 muestra un esquema de este

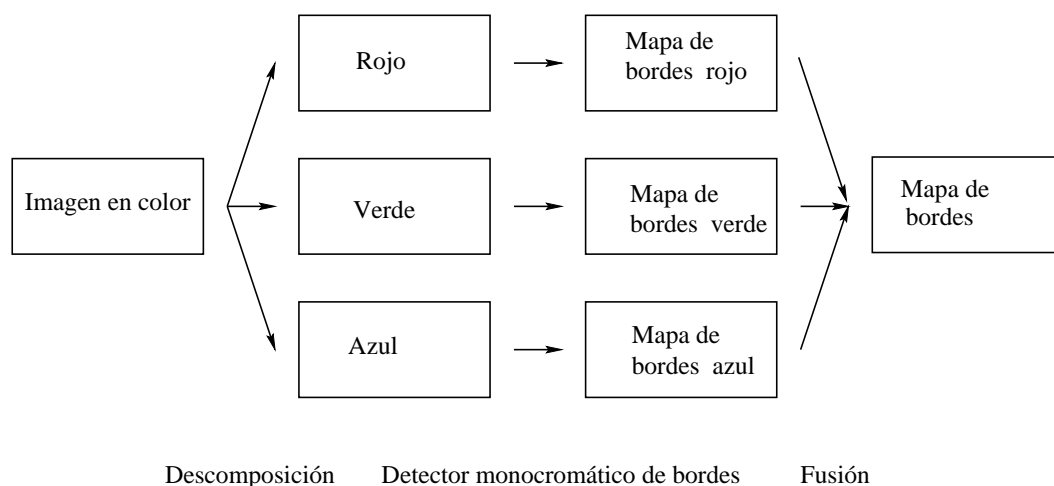


Figura 3.25: Fases de la fusión de bordes en una imagen definida en el espacio de color *RGB*.

tipo de detectores al ser aplicados a imágenes definidas sobre el espacio de color *RGB*.

Para fusionar los bordes obtenidos en cada una de las componentes de la imagen, se han propuesto diferentes alternativas:

- Fusionar los bordes mediante alguna operación lógica (Koschan 1995; Lanser y Eckstein 1992; Lanser 1993; Plataniotis y Venetsanopoulos 2000; Skarbek y Koschan 1994; Young 1986a; Zhu et al. 1999).
- Los bordes de color se obtienen a partir de los bordes obtenidos en cada uno de los planos del espacio de color que cumplan algunas restricciones de uniformidad. Esta alternativa difiere de la anterior en que las restricciones impuestas afectan a la obtención de los bordes de cada plano de color.

Nevatia (1.977) utilizó este método para extender el operador monocromático de Hueckel (1.971, 1.973) a imágenes en color, donde la restricción de uniformidad que impuso a los bordes de color fue que deberían tener la misma orientación en cada una de las componentes del espacio de color.

- Considerar la fusión de bordes como un problema de minimización (Delcroix 1988).
- Fusionar los bordes mediante un método de regularización que tiene en cuenta la curvatura de los bordes (Salinas et al. 1996).

Algunos científicos no creen que el sistema visual humano extraiga las características de una imagen en color proyectándola previamente en tres ejes independientes. Por ello, consideran poco adecuado utilizar esta extensión de las

técnicas monocromáticas para detectar bordes en imágenes en color (Ruzon y Tomasi 1999).

Otro inconveniente de esta estrategia es que no tiene en cuenta la correlación que pueda existir entre las componentes de la imagen en color, sobre todo si se utiliza el espacio de color *RGB* (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999).

Pérez y Koch (1.986) propusieron un detector muy diferente de los anteriores que también utiliza los mapas de bordes. Su detector opera sobre imágenes definidas en el espacio de color *HSI*, evitándose así la correlación de las componentes de color *RGB*. El método propuesto intenta detectar bordes provocados por cambios materiales a la vez que evita detectar bordes generados por reflejos, sombras o transparencias. El funcionamiento del detector consta de dos fases:

- Se aplica el operador de Canny (1.986) para generar el mapa de bordes en la componente de intensidad.
- Se eliminan los bordes del mapa de bordes de intensidad que no sean provocados por diferencias de tono (*hue*). Para detectar las discontinuidades de tono, se utilizó una técnica basada en los campos aleatorios de Markov para minimizar una función de energía.

Los algoritmos basados en los campos aleatorios de Markov poseen un buen comportamiento en entornos ruidosos (Jain 1981; Geman y Geman 1984; Geman 1988; Kerfoot y Bresler 1999; Modestino y Zhang 1994), pero tienen como gran inconveniente su alto coste computacional, el cual debe ser reducido mediante su implementación en paralelo (Pal y Pal 1993; Kim et al. 1998).

3.4.2.4 Detectores basados en los gradientes monocromáticos

Las técnicas monocromáticas de detección de bordes basadas en el cálculo del gradiente pueden ser extendidas a imágenes en color (o multiespectrales) (Plataniotis y Venetsanopoulos 2000; Ruzon y Tomasi 1999; Zenzo 1986; Zhu et al. 1999). El proceso consta de las siguientes fases (figura 3.26):

1. Descomposición de la imagen en sus componentes de color.
2. Estimación del gradiente monocromático en cada una de las componentes de la imagen en color.
3. Combinación de los gradientes monocromáticos para calcular el gradiente de la imagen en color.

4. Identificación de los bordes mediante umbralización. Esta fase se puede combinar con una fase previa de supresión de puntos que no son máximos locales en la dirección del gradiente, o con una fase posterior de adelgazamiento de bordes.

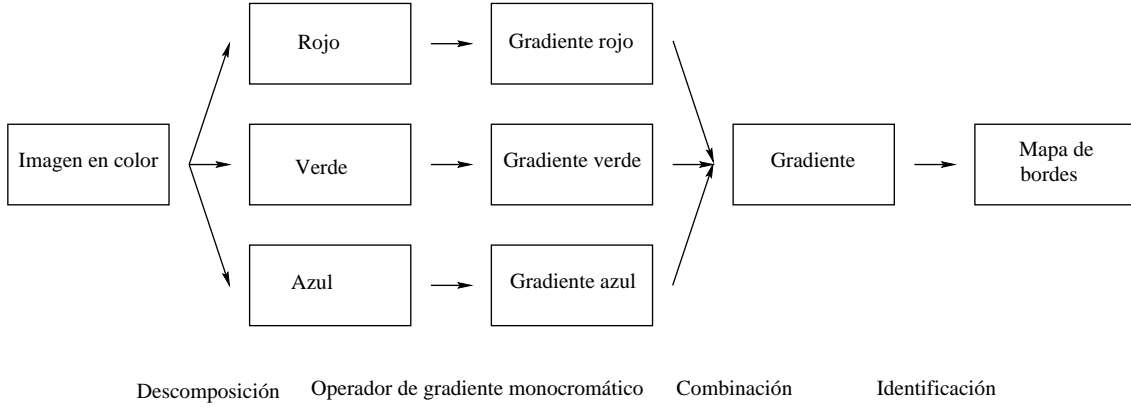


Figura 3.26: Extensión del gradiente monocromático a imágenes definidas en el espacio de color *RGB*.

El principal problema de los métodos basados en esta estrategia radica en “cómo combinar los gradientes de las componentes de color en un único gradiente” (Machuca y Philips 1983). Algunos de los posibles métodos son (Hedley y Yan 1992; Novak y Shafer 1987; Robinson 1976; Robinson 1977a; Pratt 1991; Zenzo 1986):

- Sumar los gradientes de cada una de las componentes.

$$\vec{\nabla} \vec{f}(x, y) = \vec{\nabla} R(x, y) + \vec{\nabla} G(x, y) + \vec{\nabla} B(x, y) \quad (3.60)$$

- Estimar la magnitud del gradiente como el máximo de las magnitudes de los gradientes monocromáticos.

$$\|\vec{\nabla} \vec{f}(x, y)\| = \max\{\|\vec{\nabla} R(x, y)\|, \|\vec{\nabla} G(x, y)\|, \|\vec{\nabla} B(x, y)\|\} \quad (3.61)$$

- Estimar la magnitud del gradiente como la raíz cuadrada de la suma de los cuadrados de las magnitudes de los gradientes monocromáticos.

$$\|\vec{\nabla} \vec{f}(x, y)\| = \sqrt{\|\vec{\nabla} R(x, y)\|^2 + \|\vec{\nabla} G(x, y)\|^2 + \|\vec{\nabla} B(x, y)\|^2} \quad (3.62)$$

- Estimar la magnitud del gradiente como la suma de las magnitudes de los gradientes monocromáticos.

$$\|\vec{\nabla} \vec{f}(x, y)\| = \|\vec{\nabla} R(x, y)\| + \|\vec{\nabla} G(x, y)\| + \|\vec{\nabla} B(x, y)\| \quad (3.63)$$

La utilización de los gradientes monocromáticos para estimar el gradiente de color (o multiespectral) está considerada como insatisfactoria (Zenko 1986; Zhu et al. 1999). Ello es debido a los siguientes motivos:

- No se tiene en cuenta la posible correlación entre las componentes del espacio de color.
- Si se utiliza la suma de los gradientes monocromáticos para estimar el gradiente de color (o multiespectral), puede ocurrir que no se detecten todos los bordes. Por ejemplo, considérese que dos de los gradientes monocromáticos en un punto tienen igual magnitud pero direcciones opuestas y que el otro gradiente monocromático es el vector cero, entonces el vector suma correspondiente al gradiente de color (o multiespectral) será el vector nulo y, por tanto, no se detectará ningún borde en dicho punto.
- Si se utilizan los otros tres métodos que estiman la magnitud del gradiente, el problema anterior no existe. Sin embargo, puede ocurrir que se detecten falsos bordes. Supóngase que la imagen contiene un borde
 - que presenta la misma dirección en todas las componentes
 - o que presenta direcciones diferentes, posiblemente ortogonales, en cada una de las componentes.

Todos los métodos que estiman la magnitud del gradiente asignarían el mismo valor en ambos casos, cuando deberían asignar un valor mayor a los bordes que se encuentran en la primera situación.

También se han propuesto otros detectores de bordes en color que utilizan los gradientes monocromáticos de forma diferente:

- Combinación de las estimaciones del gradiente de cada una de las componentes de color mediante un algoritmo de relajación probabilística (Macaire et al. 1996).
- Cálculo de los gradientes en el espacio de color HSV (Tsang y Tsang 1996; Tsang y Tsang 1997). Este método fue diseñado para evitar la detección de bordes provocados por reflejos especulares. El gradiente de un punto de una imagen en color es calculado como:

$$\vec{\nabla} f(x, y) = \begin{cases} \vec{\nabla} H(x, y) & \text{Si } I(x, y) \geq U \\ \vec{\nabla} V(x, y) & \text{Si } I(x, y) \leq L \\ \vec{\nabla} H(x, y) & \text{Si } U \geq I(x, y) \leq L \wedge H(x, y) \geq V(x, y) \\ \vec{\nabla} V(x, y) & \text{Si } U \geq I(x, y) \leq L \wedge H(x, y) < V(x, y) \end{cases} \quad (3.64)$$

donde U y L son dos umbrales preestablecidos e

$$I(x, y) = V(x, y)S(x, y) \quad (3.65)$$

es un índice que permite valorar la información suministrada por el tono (*hue*). Cuanto mayor sea el valor de $I(x, y)$, más fiable será la información suministrada por el tono.

- Uso de una métrica de color y de un orden en un espacio de color de escala uniforme (*USC: Uniform Scale Chromaticity space*) para extender el detector de Canny a imágenes en color (Fuertes et al. 1997). Este método fue aplicado a imágenes definidas en el espacio de color *HSI*.

3.4.3 Detectores que consideran a las imágenes en color como campos vectoriales

Se han desarrollado muchas técnicas de detección de bordes que consideran a las imágenes en color como campos vectoriales que asocian a cada punto (x, y) un vector que tiene como componentes sus valores en cada uno de los planos del espacio de color. En el caso de una imagen f definida en el espacio de color *RGB*, se asignaría a cada punto (x, y) el siguiente vector:

$$\vec{f}(x, y) = (R(x, y), G(x, y), B(x, y)) \quad (3.66)$$

Esta estrategia, que considera a la imágenes como campos vectoriales, fue sugerida inicialmente por Machuca y Phillips (1983), los cuales se preocuparon por caracterizar los bordes que no eran provocados por cambios de iluminación (intensidad) o saturación, sino por cambios de tono de color. Haciendo uso de las componentes I y Q del espacio de color YIQ, asociaron a cada punto (x, y) de la imagen un vector

$$\vec{v}(x, y) = (I(x, y), Q(x, y)) \quad (3.67)$$

donde

$$\theta(\vec{v}(x, y)) = \arctan\left(\frac{Q(x, y)}{I(x, y)}\right) \quad (3.68)$$

representa el tono de color y

$$\|\vec{v}(x, y)\| \quad (3.69)$$

su saturación.

Machuca y Phillips utilizaron el valor de $\theta(\vec{v}(x, y))$ para detectar los bordes en los puntos que tuvieran una mayor curvatura.

La utilización de las imágenes en color como campos vectoriales trata de utilizar de forma conjunta la información suministrada por cada una de las componentes del espacio de color.

Algunas de las técnicas o conceptos utilizados por los detectores de bordes en color que consideran a la imágenes en color como campos vectoriales son:

- Reducción de la dimensionalidad de la imagen mediante la proyección de los vectores sobre un eje: esta técnica ya fue descrita en el apartado 3.4.2.2.
- Ordenamiento de vectores.
- Vector gradiente multidimensional.
- Derivadas de segundo orden.
- Entropía.
- Técnicas de *clustering*.
- Histogramas.
- Teoría bayesiana.
- Polinomios ortogonales.

3.4.3.1 Detectores de bordes en color basados en el ordenamiento de vectores

Una clase importante de detectores de bordes en color está basada en estadísticos de orden. Estos métodos están inspirados en los detectores de bordes morfológicos (Feechs y Arce 1987; Lee et al. 1987) que han sido diseñados para imágenes monocromáticas. Esta clase de detectores de bordes en color se caracteriza por usar combinaciones lineales de vectores ordenados. El rendimiento y la eficiencia de este tipo de detectores de bordes depende de los coeficientes utilizados en las combinaciones lineales de los vectores ordenados (Plataniotis y Venetsanopoulos 2000; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999).

La primera etapa de los métodos basados en estadísticos de orden consiste en ordenar un conjunto de variables aleatorias en orden ascendente según un cierto criterio. Puesto que las imágenes de color son representadas mediante un espacio vectorial tridimensional, no existe ningún orden total que pueda ser definido. Sin embargo, sí es posible definir métodos de ordenamiento parcial como (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999):

- Ordenamiento marginal (*Marginal ordering*, *M-ordering*).
- Ordenamiento (agregado) reducido (*Reduced (aggregate) ordering*, *R-ordering*).
- Ordenamiento parcial (*Partial ordering*, *P-ordering*).

- Ordenamiento condicional (*Conditional ordering*, *C-ordering*).

El ordenamiento reducido *R-ordering* es el método más adecuado para el procesamiento de imágenes en color. Inicialmente se requiere una ventana W de n puntos centrada sobre cada punto de la imagen. A cada punto P_i de la ventana se le asocia un vector con las coordenadas de color $\vec{X}_i = (x_i^1, x_i^2, x_i^3)$. El método *R-ordering* utiliza una medida de distancia para asociar a cada vector \vec{X}_i un valor escalar d_i , es decir, cada vector es “reducido” a un valor numérico. A continuación son ordenados ascendentemente los valores escalares d_1, d_2, \dots, d_n , obteniéndose la permutación

$$d^{(1)} \leq d^{(2)} \leq \dots \leq d^{(n)} \quad (3.70)$$

Esta ordenación de los valores escalares permite reordenar los vectores originales

$$\vec{X}^{(1)} \leq \vec{X}^{(2)} \leq \dots \leq \vec{X}^{(n)} \quad (3.71)$$

En esta secuencia ordenada, el vector $\vec{X}^{(1)}$ es “el vector mediana” mientras que los vectores que aparecen en las posiciones más altas son denominados “*outliers*” y se caracterizan porque son aquéllos que más divergen.

El método *R-ordering* también suele llamarse “ordenación agregada” debido a que a veces se utiliza como distancia métrica para reordenar los vectores “la distancia agregada” de un vector \vec{X}_i respecto a los demás vectores:

$$d_i = \sum_{k=1}^n \|\vec{X}_i - \vec{X}_k\|, \quad k \neq i \quad (3.72)$$

Una alternativa a la distancia agregada es usar la distancia de cada vector al vector asociado al punto central de la ventana W (Puentes-Rojas 1997).

Generalmente, la distancia entre los vectores se define en función de la norma L_2 o distancia euclídea:

$$\|\vec{X}_i - \vec{X}_j\|_2 = \sqrt{\sum_{k=1}^p (x_i^k - x_j^k)^2} \quad (3.73)$$

donde⁵ $\vec{X}_i = (x_i^1, x_i^2, \dots, x_i^p)$ y $\vec{X}_j = (x_j^1, x_j^2, \dots, x_j^p)$.

Sin embargo, la distancia euclídea no es la única distancia que se puede utilizar para el procesamiento de imágenes digitales. Otras distancias que se han propuesto son⁶ (Duda y Hart 1973; Plataniotis y Venetsanopoulos 2000; Sonka et al. 1993):

⁵En el caso de las imágenes en color, $p = 3$.

⁶La distancia de Canberra y el coeficiente de Czekanowski sólo se pueden aplicar a vectores con componentes no negativas, lo cual se verifica en los espacios de color (Plataniotis y Venetsanopoulos 2000).

- La norma L_1 o distancia de “la ciudad de bloques” (*city block - distance*) o de “Manhattan”:

$$||\vec{X}_i - \vec{X}_j||_1 = \sum_{k=1}^p |x_i^k - x_j^k| \quad (3.74)$$

- La norma L_∞ o distancia de “ajedrez” (*chess-board distance*):

$$||\vec{X}_i - \vec{X}_j||_\infty = \max_k \{|x_i^k - x_j^k|\}, \quad k = 1, 2, \dots, p \quad (3.75)$$

- Distancia angular:

$$||\vec{X}_i - \vec{X}_j||_a = \arccos \left(\frac{\vec{X}_i \cdot \vec{X}_j}{||\vec{X}_i|| ||\vec{X}_j||} \right) \quad (3.76)$$

donde $||\vec{X}||$ denota la magnitud del vector \vec{X} .

- La distancia de Canberra:

$$||\vec{X}_i - \vec{X}_j||_c = \sum_{k=1}^p \frac{|x_i^k - x_j^k|}{|x_i^k + x_j^k|} \quad (3.77)$$

- El coeficiente de Czekanowski:

$$||\vec{X}_i - \vec{X}_j||_z = 1 - \frac{2 \sum_{k=1}^p \min(x_i^k, x_j^k)}{\sum_{k=1}^p (x_i^k + x_j^k)} \quad (3.78)$$

Detectores de bordes en color basados en el método *R-ordering*.

El detector de bordes más simple que se puede definir utilizando el método de ordenamiento *R-ordering* es el detector bordes en color *VR* (*Vector range* o “rango de los vectores”) (Plataniotis y Venetsanopoulos 2000; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999):

$$VR = ||\vec{X}^{(n)} - \vec{X}^{(1)}|| \quad (3.79)$$

VR representa la desviación que hay entre el vector *outlier* de mayor rango y el vector mediana en la ventana W . El valor VR obtenido en cada ventana W es asignado al punto central de la misma. Cuando la distancia entre los vectores de W sea cercana, el valor de VR será pequeño, lo que significa que W está situada sobre una región uniforme, es decir, con poca variación. Sin embargo, cuando hay una presencia de un borde en W , la respuesta de VR será mayor, ya que $\vec{X}^{(1)}$ pertenecerá a un lado del borde en el que los niveles de color de los puntos son muy diferentes a los del otro lado del borde, donde está situado $\vec{X}^{(n)}$. Por consiguiente, los bordes pueden ser obtenidos umbralizando los valores asignados por VR a cada punto.

Este método es simple y eficiente, pero presenta como principal desventaja el hecho de que es muy sensible al ruido, especialmente al ruido de tipo impulsivo, ya que normalmente los vectores *outliers* corresponden a puntos contaminados con ruido.

Para solucionar este problema se ha propuesto una serie de medidas de dispersión que ofrecen unos cálculos más robustos en presencia de ruido. Una de estas medidas es el denominado *VDED* (*Vector dispersion edge detector* o “detector de bordes basado en la dispersión de vectores”) que utiliza combinaciones lineales de los vectores ordenados (Trahanias y Venetsanopoulos 1993):

$$\begin{aligned} VDED &= OSO \left(\left\| \sum_{i=1}^n \alpha_{i1} \vec{X}^{(i)} \right\|, \left\| \sum_{i=1}^n \alpha_{i2} \vec{X}^{(i)} \right\|, \dots, \left\| \sum_{i=1}^n \alpha_{ik} \vec{X}^{(i)} \right\| \right) \\ &= OSO_j \left(\left\| \sum_{i=1}^n \alpha_{ij} \vec{X}^{(i)} \right\| \right), j = 1, 2, \dots, k \end{aligned} \quad (3.80)$$

donde $\|\cdot\|$ es una norma apropiada y *OSO* (*order statistic operator*) representa a un operador estadístico basado en la ordenación. A partir de la elección de los coeficientes α_{ij} y de *OSO* se pueden definir diferentes detectores de bordes en color. Un ejemplo simple es

$$VDED = \left\| \sum_{i=1}^n \alpha_i \vec{X}^{(i)} \right\| \quad (3.81)$$

El detector *VR* es un caso especial del detector definido en 3.81, ya que se obtendría si se eligieran los siguientes coeficientes: $\alpha_1 = -1$, $\alpha_n = 1$ y $\alpha_i = 0, \forall i \in \{2, \dots, n-1\}$.

La elección del operador *OSO* y de los coeficientes α_{ij} que permita obtener el detector de bordes más eficaz presenta una gran dificultad desde un punto de vista matemático, por lo que se opta por un método alternativo que consiste en establecer una serie de requisitos que todo detector de bordes debe cumplir:

- Debe ser insensible al ruido de “cola larga” o *long-tailed* (impulsivo o doble exponencial) y de “cola corta” o *short-tailed* (gaussiano).
- Debe responder apropiadamente a bordes en forma de rampa o pendiente, asignando un valor mayor a los puntos del borde que a los situados alrededor.

Puesto que el ruido de tipo impulsivo suele aparecer en forma de *outliers* en la secuencia ordenada, se puede definir un detector que calcule k valores de *VR* en vez de uno solo y después tomar el mínimo de ellos:

$$VR_{min} = \min_j \{ \left\| \vec{X}^{(n-j+1)} - \vec{X}^{(1)} \right\| \}, j = 1, 2, \dots, k, k < n \quad (3.82)$$

El efecto del operador VR_{min} es ignorar hasta $k - 1$ falsos impulsos debidos a puntos ruidosos aislados. Para elegir el valor de k hay que tener en cuenta el valor de n que indica el tamaño de la ventana W . Si se utiliza una ventana cuadrada de dimensión $m \times m$ entonces $k \leq m$. Sin embargo, no existe ningún cálculo matemático que permita obtener un valor de k adecuado, por lo que debe obtenerse mediante una apreciación subjetiva. De todos modos, el valor de k puede calcularse intuitivamente ya que representa el número de puntos que residen en el lado más pequeño de un borde cuando la ventana W se centra sobre un punto perteneciente al borde.

Por otro lado, el ruido de “cola corta” puede evitarse en cierta medida si se sustituye el vector mediana $\vec{X}^{(1)}$ por el vector denominado “ $\overrightarrow{V\alpha TM}$ ” (*Vector α -trimmed mean* o “media α -ajustada de los vectores”):

$$\overrightarrow{V\alpha TM} = \sum_{i=1}^l \left(\frac{\vec{X}^{(i)}}{l} \right) \quad (3.83)$$

Combinando las ecuaciones 3.82 y 3.83 se obtiene un nuevo detector de bordes denominado “*MVD*” (*Minimum vector dispersion* o “dispersión de vectores mínima”):

$$MVD = \min_j \left\{ \left\| \vec{X}^{(n-j+1)} - \sum_{i=1}^l \frac{\vec{X}^{(i)}}{l} \right\| \right\} \quad j = 1, 2, \dots, k. \quad k, l < n \quad (3.84)$$

El valor de l no puede ser definido formalmente como se hizo con el parámetro k , pero tiene una íntima relación con k en el sentido de que l expresa el número de puntos que pertenecen al lado mayor de un borde cuando W está centrada en un punto del borde.

El detector *MVD* tiene un mejor rendimiento en presencia de ruido puesto que la operación de minimización lo hace insensible al ruido de “cola larga” y la operación de promediado lo hace insensible al ruido de “cola corta”. También se debe destacar que el detector *MVD* responde a los bordes de rampa asignando un valor mayor a los puntos pertenecientes al borde que a los circundantes.

El operador *MVD* se puede simplificar si se tiene en cuenta que los vectores asociados a los posibles puntos contaminados con ruido impulsivo van a ocupar los lugares de mayor rango en la serie ordenada de vectores $(\vec{X}^{(n-k+2)}, \vec{X}^{(n-k+3)}, \dots, \vec{X}^{(n)})$. Puesto que la distancia entre estos vectores y el resto de vectores es grande, la ecuación 3.84 puede ser reducida a:

$$MVD = \left\| \vec{X}^{(n-k+1)} - \sum_{i=1}^l \frac{\vec{X}^{(i)}}{l} \right\| \quad (3.85)$$

Este detector no utiliza ninguno de los posibles puntos ruidosos y, por tanto, no se ve afectado por este tipo de ruido impulsivo. Además, el detector continúa

siendo robusto al ruido gaussiano debido a la operación de promediado de l puntos.

Se puede definir una variante del detector *VDED* que utiliza “el filtro adaptativo del vecino más próximo” (*adaptive nearest-neighbor (NN) filter*). Este detector tiene en cuenta las características locales de la imagen y en vez de utilizar unos coeficientes constantes, hace uso de unos coeficientes específicos para cada ventana W . Dichos coeficientes son determinados por una función de ponderación. El operador es definido como la distancia entre el *outlier* y la suma ponderada de todos los vectores:

$$NNVR = \left\| \vec{X}^{(n)} - \sum_{i=1}^n w_i \vec{X}^{(i)} \right\| \quad (3.86)$$

Los valores de los coeficientes w_i deben ser determinados para cada ventana W , pero han de cumplir dos restricciones:

- Deben ser positivos: $w_i \geq 0, \forall i = 1, 2, \dots, n$
- Estar normalizados:

$$\sum_{i=1}^n w_i = 1 \quad (3.87)$$

Para atenuar el ruido, se deben asignar coeficientes pequeños a los vectores de mayor rango (*outliers*). Una posible función puede ser definida como

$$w_i = \frac{d^{(n)} - d^{(i)}}{n \cdot d^{(n)} - \sum_{j=1}^n d^{(j)}} \quad (3.88)$$

En el caso de que la ventana W esté situada sobre una región uniforme, el denominador de 3.88 puede ser cero, con lo que la función no puede ser utilizada. No obstante, la uniformidad de la región pone de manifiesto que no contiene ningún borde, con lo que el valor de *NNVR* debe ser cero.

Los operadores *MVD* y *NNVR* pueden ser combinados para obtener un detector que posea un mejor rendimiento ante el ruido impulsivo:

$$NNMVD = \min_j \left\{ \left\| \vec{X}^{(n-j+1)} - \sum_{i=1}^n w_i \vec{X}^{(i)} \right\| \right\}, \quad j = 1, 2, \dots, k. \quad k < n \quad (3.89)$$

Todos estos detectores basados en el ordenamiento de vectores localizan los puntos de los bordes mediante la umbralización de los valores asignados a los puntos de la imagen en color.

3.4.3.2 Detectores de bordes en color basados en el vector gradiente

El concepto de vector gradiente ha sido utilizado para diseñar algoritmos de detección de bordes en color. Estos detectores tratan de aplicar a imágenes en color una estrategia similar a la utilizada por los detectores de bordes monocromáticos basados en el gradiente.

Se han propuesto diferentes métodos para calcular el vector gradiente en un punto de una imagen en color:

- Métodos que usan distancias métricas para estimar el vector gradiente.
- Gradiente multidimensional.
- Operadores vectoriales direccionales.
- Operadores basados en la diferencia de vectores.

Uso de distancias métricas para estimar el vector gradiente

Se han diseñado algunos métodos de detección de bordes en color que utilizan una distancia métrica para estimar la magnitud y la dirección del gradiente:

- Uno de los métodos más sencillos para estimar el vector gradiente en un punto de una imagen en color hace uso de la distancia euclídea (norma L_2) y de una ventana W de 3×3 puntos (Plataniotis y Venetsanopoulos 2000; Shiozaki 1986; Zhu et al. 1999). El método consiste en calcular las distancias euclídeas entre el vector de color asociado al punto central ($\vec{f}(x_0, y_0)$) y los vectores de color asociados a sus ocho vecinos ($\vec{f}(x_i, y_i), i = 1, \dots, 8$). Estas ocho distancias son utilizadas para estimar la magnitud del gradiente en el punto central como

$$||\vec{\nabla} f(x_0, y_0)|| = \max_{i=1, \dots, 8} \{||\vec{f}(x_i, y_i) - \vec{f}(x_0, y_0)||\} \quad (3.90)$$

La dirección del gradiente es indicada por el vector que une el punto central con el punto en el que se ha alcanzado el máximo.

- Se pueden obtener variantes del método anterior utilizando otras distancias métricas, como la distancia de la ciudad de los bloques (norma L_1), la distancia de ajedrez (norma L_∞) o la distancia angular (Plataniotis y Venetsanopoulos 2000; Wesolkowski y Jernigan 1999; Zhu et al. 1999). También se ha propuesto un método que combina la distancia euclídea y la distancia angular (Wesolkowski y Jernigan 1999).

- La magnitud y la dirección del vector gradiente también han sido estimadas utilizando un detector “borroso” (*fuzzy*) que combina una función normalizada de contraste de tono (*normalized hue contrast*) definida en el espacio de color *HSI* y la distancia euclídea en el espacio de color *RGB* (Moghadamzadeh y Bourbakis 1997).
- También se ha utilizado un operador de brújula (*compass operator*) junto con la “distancia de La Tierra” (*Earth mover’s distance*) (Rubner et al. 1998) definida en el espacio de color CIE Lab para estimar un vector similar al vector gradiente (Ruzon y Tomasi 1999).

Gradiente multidimensional

Para resolver el problema de la combinación de los gradientes monocromáticos (Machuca y Philips 1983), Di Zenzo (1.986) propuso un nuevo método, basado en el uso de los operadores direccionales, que permite calcular el gradiente de una imagen multidimensional.

Los operadores direccionales horizontal y vertical en el punto (x, y) de la imagen f se definen como:

$$\vec{u}(x, y) = \frac{\partial R}{\partial x}(x, y)\vec{r} + \frac{\partial G}{\partial x}(x, y)\vec{g} + \frac{\partial B}{\partial x}(x, y)\vec{b} \quad (3.91)$$

y

$$\vec{v}(x, y) = \frac{\partial R}{\partial y}(x, y)\vec{r} + \frac{\partial G}{\partial y}(x, y)\vec{g} + \frac{\partial B}{\partial y}(x, y)\vec{b} \quad (3.92)$$

donde \vec{r} , \vec{g} y \vec{b} son los vectores unitarios de los ejes R , G y B , respectivamente.

Estos operadores direccionales permiten definir, a su vez, las siguientes cuatro componentes de un tensor simétrico

$$g_{xx}(x, y) = \vec{u}(x, y) \cdot \vec{u}(x, y) = \left(\frac{\partial R}{\partial x}(x, y) \right)^2 + \left(\frac{\partial G}{\partial x}(x, y) \right)^2 + \left(\frac{\partial B}{\partial x}(x, y) \right)^2 \quad (3.93)$$

$$g_{yy}(x, y) = \vec{v}(x, y) \cdot \vec{v}(x, y) = \left(\frac{\partial R}{\partial y}(x, y) \right)^2 + \left(\frac{\partial G}{\partial y}(x, y) \right)^2 + \left(\frac{\partial B}{\partial y}(x, y) \right)^2 \quad (3.94)$$

$$g_{xy}(x, y) = g_{yx}(x, y) = \vec{u}(x, y) \cdot \vec{v}(x, y) = \frac{\partial R}{\partial x}(x, y) \frac{\partial R}{\partial y}(x, y) + \frac{\partial G}{\partial x}(x, y) \frac{\partial G}{\partial y}(x, y) + \frac{\partial B}{\partial x}(x, y) \frac{\partial B}{\partial y}(x, y) \quad (3.95)$$

Se puede utilizar este tensor para obtener la dirección y la magnitud del vector gradiente de \vec{f} en cada punto (x, y) , para lo cual se ha de calcular inicialmente la distancia euclídea (norma L_2) entre dos vectores próximos $\vec{f}(x, y)$ y

$\vec{f}(x + \theta, y + \theta)$. Esta distancia puede ser expresada en función de los operadores direccionales como

$$df^2 = ||\vec{u}(x, y)||^2 \cos^2(\theta) + 2 \vec{u}(x, y) \cdot \vec{v}(x, y) \cos(\theta) \sin(\theta) + ||\vec{v}(x, y)||^2 \sin^2(\theta) \quad (3.96)$$

o en función de las componentes del tensor como

$$df^2 = F(\theta) = g_{xx}(x, y) \cos^2(\theta) + 2g_{xy}(x, y) \cos(\theta) \sin(\theta) + g_{yy}(x, y) \sin^2(\theta) \quad (3.97)$$

Teniendo en cuenta que

$$\sin^2(\theta) = \frac{1}{2}(1 - \cos(2\theta)) \quad (3.98)$$

$$\cos^2(\theta) = \frac{1}{2}(1 + \cos(2\theta)) \quad (3.99)$$

$$\sin(\theta) \cos(\theta) = \frac{1}{2} \sin(2\theta) \quad (3.100)$$

se obtiene la siguiente función de θ

$$\begin{aligned} df^2 &= F(\theta) \\ &= \frac{1}{2}(g_{xx}(x, y) + g_{yy}(x, y)) + (g_{xx}(x, y) - g_{yy}(x, y)) \cos(2\theta) + 2g_{xy}(x, y) \sin(2\theta) \end{aligned} \quad (3.101)$$

Para calcular la dirección y la magnitud del gradiente se ha de determinar el valor de θ que maximice la función 3.101, para lo cual se han de obtener las soluciones de

$$\frac{\partial F(\theta)}{\partial \theta} = 0 \quad (3.102)$$

Esta ecuación tiene las siguientes soluciones

$$\theta = \frac{1}{2} \arctan \left(\frac{2g_{xy}(x, y)}{g_{xx}(x, y) - g_{yy}(x, y)} \right) \quad (3.103)$$

$$\text{y } \theta \pm \frac{\pi}{2}.$$

Entre estas tres posibles soluciones, se toma aquella que maximice $F(\theta)$. La solución obtenida θ_0 es la estimación de la dirección del gradiente y la raíz cuadrada del máximo obtenido es la estimación de la magnitud del gradiente.

Para identificar los bordes, se han utilizado los siguientes métodos:

- Umbralizar el valor de $\sqrt{F(\theta_0)}$ (Plataniotis y Venetsanopoulos 2000; Zenzo 1986; Zugaj y Lattuat 1998) o de $F(\theta_0)$ (Chapron 1992).
- Localizar los máximos locales en la dirección del gradiente (Drewniok 1994; Puentes-Rojas 1997).

Para estimar los valores de los gradientes elementales en los planos de rojo (R), verde (G) y azul (B) ($\frac{\partial R}{\partial x}, \frac{\partial R}{\partial y}, \frac{\partial G}{\partial x}, \frac{\partial G}{\partial y}, \frac{\partial B}{\partial x}, \frac{\partial B}{\partial y}$), Di Zenzo (1.986) propuso utilizar el operador de Sobel, mientras que Drewniok (1.994) usó el operador de Canny (1.986) para aplicar este método a imágenes multispectrales obtenidas vía satélite. Chapron (1.992) sugirió utilizar el método de Deriche (1.987), el cual también está basado en el detector de Canny.

Este método de cálculo del gradiente multidimensional presenta algunas indeterminaciones en la dirección del gradiente cuando $g_{xx} = g_{yy}$ y $g_{xy} = 0$ o cuando los valores de g_{xx} y g_{yy} están muy próximos. Para resolver esta ambigüedad se ha propuesto un método similar que utiliza la matriz con las primeras derivadas de la imagen en color (Lee y Cok 1991; Novak y Shafer 1987; Saber et al. 1997):

$$D(x, y) = \begin{bmatrix} \frac{\partial R}{\partial x}(x, y) & \frac{\partial G}{\partial x}(x, y) & \frac{\partial B}{\partial x}(x, y) \\ \frac{\partial R}{\partial y}(x, y) & \frac{\partial G}{\partial y}(x, y) & \frac{\partial B}{\partial y}(x, y) \end{bmatrix} \quad (3.104)$$

La matriz D se denomina “matriz jacobiana” de \vec{f} y permite definir una distancia de color d entre $\vec{f}(x, y)$ y $\vec{f}(x + n_1, y + n_2)$, que se corresponde con los cambios en los atributos de la imagen,

$$d = \sqrt{\vec{n} D D^T \vec{n}^T} \quad (3.105)$$

donde $\vec{n} = (n_1, n_2)$ es un vector unitario.

Para maximizar el valor de d es necesario calcular el autovalor más grande de la matriz simétrica

$$D(x, y) D^T(x, y) = \begin{bmatrix} g_{xx}(x, y) & g_{xy}(x, y) \\ g_{xy}(x, y) & g_{yy}(x, y) \end{bmatrix} \quad (3.106)$$

El autovalor más grande puede ser expresado por

$$\lambda(x, y) = \frac{1}{2} \left[g_{xx}(x, y) + g_{yy}(x, y) + \sqrt{(g_{xx}(x, y) + g_{yy}(x, y))^2 - 4(g_{xx}(x, y)g_{yy}(x, y) - g_{xy}^2(x, y))} \right] \quad (3.107)$$

La estimación de la magnitud del gradiente es calculada como

$$\|\vec{\nabla} f(x, y)\| = \sqrt{\lambda(x, y)} \quad (3.108)$$

La estimación de la dirección del gradiente la indica el autovector asociado al autovalor $\lambda(x, y)$, es decir,

$$\vec{v}_1(x, y) = (g_{xy}(x, y), \lambda(x, y) - g_{xx}(x, y)) \quad (3.109)$$

o

$$\vec{v}_2(x, y) = (\lambda(x, y) - g_{yy}(x, y), g_{xy}(x, y)) \quad (3.110)$$

si el vector $\vec{v}_1(x, y)$ es un vector nulo.

Este método, a diferencia del anterior, no necesita ninguna fase de maximización, ya que el valor de la estimación de la magnitud del gradiente se obtiene directamente de $\lambda(x, y)$. Sin embargo, pueden existir dificultades en la estimación de la dirección del gradiente cuando los vectores $\vec{v}_1(x, y)$ y $\vec{v}_2(x, y)$ son nulos. Esto ocurre cuando $g_{xx}(x, y) = g_{yy}(x, y)$ y $g_{xy}(x, y) = 0$.

El gradiente multiespectral permite extraer más información de la imagen en color que los métodos basados en la extensión del gradiente monocromático. Ello es debido a que el gradiente multiespectral considera la naturaleza vectorial de la imagen en color (Platanotis y Venetsanopoulos 2000; Zhu et al. 1999).

Los métodos basados en el gradiente son muy sensibles a pequeñas variaciones de textura (Trahanias y Venetsanopoulos 1996), lo cual puede dificultar la identificación de objetos reales. Además son sensibles a los ruidos de tipo gaussiano e impulsivo (Platanotis y Venetsanopoulos 2000; Zhu et al. 1999).

Operadores vectoriales direccionales

El vector gradiente de una imagen en color puede ser estimado mediante el uso de operadores vectoriales direccionales (Li y Wu 1997; Scharcanski y Venetsanopoulos 1997; Platanotis y Venetsanopoulos 2000; Zhu et al. 1999). Estos operadores vectoriales direccionales pueden ser considerados como generalizaciones de los operadores básicos, como el operador de Prewitt (1970):

$$\Delta H = \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad \Delta V = \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.111)$$

Los operadores direccionales horizontales y verticales poseen una componente positiva y otra negativa. Para operadores de tamaño $(2w + 1) \times (2w + 1)$, la configuración es la siguiente:

$$\Delta H = [H_- \ 0 \ H_+], \quad \Delta V = \begin{bmatrix} V_- \\ 0 \\ V_+ \end{bmatrix} \quad (3.112)$$

donde el parámetro w es número entero positivo. Estas componentes negativas y positivas H_- , H_+ , V_- y V_+ son denominadas “núcleos de convolución” (*convolution kernels*) que generan vectores que se corresponden con “el promedio local de los colores”. Para estimar el gradiente en el punto (x_0, y_0) , las salidas de estas

componentes son calculadas mediante las siguientes expresiones:

$$\vec{H}_+(x_0, y_0) = \frac{1}{w(2w+1)} \sum_{y=y_0-w}^{y_0+w} \sum_{x=x_0+1}^{x_0+w} \vec{f}(x, y) \quad (3.113)$$

$$\vec{H}_-(x_0, y_0) = \frac{1}{w(2w+1)} \sum_{y=y_0-w}^{y_0+w} \sum_{x=x_0-1}^{x_0-w} \vec{f}(x, y) \quad (3.114)$$

$$\vec{V}_+(x_0, y_0) = \frac{1}{w(2w+1)} \sum_{y=y_0+1}^{y_0+w} \sum_{x=x_0-w}^{x_0+w} \vec{f}(x, y) \quad (3.115)$$

$$\vec{V}_-(x_0, y_0) = \frac{1}{w(2w+1)} \sum_{y=y_0-1}^{y_0-w} \sum_{x=x_0-w}^{x_0+w} \vec{f}(x, y) \quad (3.116)$$

donde $\vec{f}(x, y)$ representa el color $(R(x, y), G(x, y), B(x, y))$ en el punto de la imagen (x, y) .

Para estimar las variaciones locales en las direcciones horizontal y vertical, se calculan las siguientes diferencias de vectores:

$$\Delta\vec{H}(x_0, y_0) = \vec{H}_+(x_0, y_0) - \vec{H}_-(x_0, y_0) \quad (3.117)$$

$$\Delta\vec{V}(x_0, y_0) = \vec{V}_+(x_0, y_0) - \vec{V}_-(x_0, y_0) \quad (3.118)$$

Los valores escalares $\|\Delta\vec{H}(x_0, y_0)\|$ y $\|\Delta\vec{V}(x_0, y_0)\|$ reflejan las variaciones en las direcciones horizontal y vertical, respectivamente. Los cambios locales en los canales o planos de color no pueden obtenerse mediante la suma de las componentes de $\Delta\vec{H}(x_0, y_0)$ y $\Delta\vec{V}(x_0, y_0)$ debido a que puede producirse una mutua compensación de sus valores en algunas situaciones, como, por ejemplo, cuando variaciones del mismo contraste se encuentren en direcciones opuestas. No obstante, se puede estimar la variación local del color considerando que las componentes son independientes y calculando la magnitud y la dirección del vector resultante:

$$\|\vec{\nabla}f(x_0, y_0)\| = \sqrt{\|\Delta\vec{H}(x_0, y_0)\|^2 + \|\Delta\vec{V}(x_0, y_0)\|^2} \quad (3.119)$$

$$\theta = \arctan \left[\frac{\Delta V'(x_0, y_0)}{\Delta H'(x_0, y_0)} \right] + k\pi \quad (3.120)$$

donde k es un entero y

$$\Delta V'(x_0, y_0) = \begin{cases} \|\Delta\vec{V}(x_0, y_0)\| & \text{Si } \|\vec{V}_+(x_0, y_0)\| \geq \|\vec{V}_-(x_0, y_0)\| \\ -\|\Delta\vec{V}(x_0, y_0)\| & \text{En otro caso} \end{cases} \quad (3.121)$$

$$\Delta H'(x_0, y_0) = \begin{cases} \|\Delta \vec{H}(x_0, y_0)\| & \text{Si } \|\vec{H}_+(x_0, y_0)\| \geq \|\vec{H}_-(x_0, y_0)\| \\ -\|\Delta \vec{H}(x_0, y_0)\| & \text{En otro caso} \end{cases} \quad (3.122)$$

donde $\|\cdot\|$ representa la norma euclídea.

Este planteamiento considera que el contraste de color no tiene signo. Sin embargo, para obtener la dirección de máximo contraste, se puede aplicar un convenio que permita atribuir signos a las cantidades $\Delta V'(x_0, y_0)$ y $\Delta H'(x_0, y_0)$ definidas en 3.121 y 3.122: estas cantidades son consideradas positivas si la luminancia incrementa en la dirección positiva de los ejes del sistema de coordenadas de la imagen. Las cantidades de luminancia son estimadas por las magnitudes de $\|\vec{H}_+(x_0, y_0)\|$, $\|\vec{H}_-(x_0, y_0)\|$, $\|\vec{V}_+(x_0, y_0)\|$ y $\|\vec{V}_-(x_0, y_0)\|$. Generalmente, la luminancia ha sido estimada por la cantidad de luminancia definida por

$$\|\vec{f}(x, y)\|_1 = R(x, y) + G(x, y) + B(x, y) \quad (3.123)$$

o por (Healey 1992)

$$\|\vec{f}(x, y)\|_2 = \sqrt{R(x, y)^2 + G(x, y)^2 + B(x, y)^2} \quad (3.124)$$

Otra posibilidad consiste en considerar el contraste local con respecto a una referencia en vez de la cantidad de luminancia. Sin embargo, esta posibilidad podría presentar algunas ambigüedades, como, por ejemplo, la que se presentaría en un borde vertical del tipo rampa donde $\|\vec{H}_-(x_0, y_0) - \vec{f}(x_0, y_0)\| = \|\vec{H}_+(x_0, y_0) - \vec{f}(x_0, y_0)\|$, lo que provocaría que $\Delta H'(x_0, y_0)$ tuviera signo positivo independientemente del signo de la pendiente de la rampa.

El parámetro w de este tipo de detectores influye en la atenuación del ruido y en los bordes bien marcados. Esto se produce porque hay un efecto de suavización (“paso bajo”) que depende del tamaño de la máscara de convolución.

Diferencia de vectores

Los detectores basados en la diferencia de vectores pueden ser considerados como detectores basados en las derivadas de primer orden. Esta clase de detectores asocia a cada punto un vector en el espacio de color RGB y después obtiene un gradiente en cada una de las cuatro posibles direcciones ($0^\circ, 45^\circ, 90^\circ$ y 135°). El gradiente que tenga mayor magnitud es asignado a cada punto. Por último, una fase final de umbralización permite localizar los puntos de los bordes (Plataniotis et al. 1997; Skarbek y Koschan 1994; Wesolkowski y Jernigan 1999; Zhu et al. 1999).

Los gradientes y sus correspondientes magnitudes son definidos como

$$\|\vec{\nabla} \vec{f}_{0^\circ}\| = \|\vec{Y}_{0^\circ} - \vec{X}_{0^\circ}\| \quad (3.125)$$

$$||\vec{\nabla} f_{45^\circ}|| = ||\vec{Y}_{45^\circ} - \vec{X}_{45^\circ}|| \quad (3.126)$$

$$||\vec{\nabla} f_{90^\circ}|| = ||\vec{Y}_{90^\circ} - \vec{X}_{90^\circ}|| \quad (3.127)$$

$$||\vec{\nabla} f_{135^\circ}|| = ||\vec{Y}_{135^\circ} - \vec{X}_{135^\circ}|| \quad (3.128)$$

donde $|| \cdot ||$ representa la norma L_2 .

El operador DV “vector diferencia” (*difference vector*) se obtiene calculando:

$$DV = \max\{||\vec{\nabla} f_{0^\circ}||, ||\vec{\nabla} f_{45^\circ}||, ||\vec{\nabla} f_{90^\circ}||, ||\vec{\nabla} f_{135^\circ}||\} \quad (3.129)$$

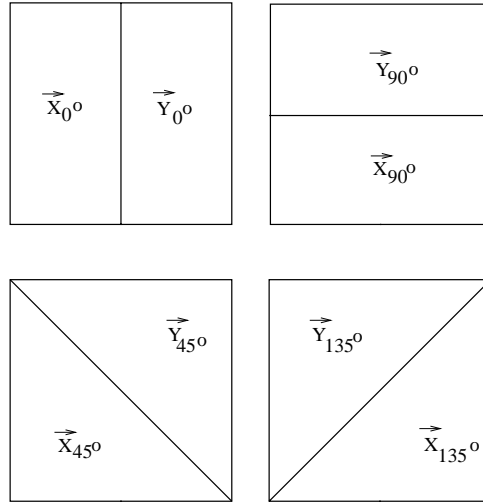


Figura 3.27: Configuraciones para la obtención de los gradientes.

Se pueden definir diferentes variantes del operador DV dependiendo del modo de obtener los gradientes. El método más simple utiliza una ventana de 3×3 puntos. A cada punto $P = (x, y)$ de la ventana se le asocia un vector $\vec{v}(x, y)$. Los vectores requeridos para calcular los cuatro gradientes correspondientes al punto central $P = (x_0, y_0)$ se definen como

$$\vec{X}_{0^\circ} = \vec{v}(x_0 - 1, y_0), \quad \vec{Y}_{0^\circ} = \vec{v}(x_0 + 1, y_0) \quad (3.130)$$

$$\vec{X}_{45^\circ} = \vec{v}(x_0 - 1, y_0 + 1), \quad \vec{Y}_{45^\circ} = \vec{v}(x_0 + 1, y_0 - 1) \quad (3.131)$$

$$\vec{X}_{90^\circ} = \vec{v}(x_0, y_0 - 1), \quad \vec{Y}_{135^\circ} = \vec{v}(x_0, y_0 + 1) \quad (3.132)$$

$$\vec{X}_{135^\circ} = \vec{v}(x_0 + 1, y_0 + 1), \quad \vec{Y}_{135^\circ} = \vec{v}(x_0 - 1, y_0 - 1) \quad (3.133)$$

El operador DV es sensible a los ruidos de tipo impulsivo y gaussiano. Sin embargo se pueden definir variantes de DV que tienen un rendimiento más robusto en presencia de ruido. Estas variantes necesitan utilizar ventanas W de un tamaño superior a 3×3 puntos. Generalmente se suelen escoger ventanas de

5×5 puntos para que la complejidad computacional no aumente considerablemente. Además, el uso de ventanas de mayor tamaño puede que impida reflejar adecuadamente las características locales de cada región. Cada ventana de $n \times n$ puntos ($n = 2k + 1, k = 2, 3, \dots$) es dividida en dos subventanas de $N = \frac{n^2-1}{2}$ puntos. La figura 3.27 muestra las disposiciones de las subventanas para cada una de las direcciones. Para poder definir los gradientes de las ecuaciones 3.125-3.128 se necesita determinar un vector para cada subventana de cada dirección. Cada uno de estos vectores puede ser obtenido mediante un filtro que se aplicaría a los vectores de cada una de las subventanas:

$$\vec{X}_{d^o} = f(\vec{v}_{d^o,1}^{sub_1}, \vec{v}_{d^o,2}^{sub_1}, \dots, \vec{v}_{d^o,N}^{sub_1}) \quad (3.134)$$

$$\vec{Y}_{d^o} = f(\vec{v}_{d^o,1}^{sub_2}, \vec{v}_{d^o,2}^{sub_2}, \dots, \vec{v}_{d^o,N}^{sub_2}) \quad (3.135)$$

donde $d = 0, 45, 90, 135$ y $\vec{v}_{d^o,i}^{sub_j}$ ($i = 1, 2, \dots, N; j = 1, 2$) es el vector asociado al punto P_i de las subventana j en la dirección d^o .

Dependiendo del tipo de ruido que se desee atenuar, se pueden utilizar diferentes filtros. A continuación se van a describir cuatro filtros basados en estadísticos de orden:

- Filtro del vector de la mediana: los vectores $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N$ de cada subventana son reordenados según el método *R-ordering* obteniéndose

$$\vec{v}^{(1)} \leq \vec{v}^{(2)} \leq \dots \leq \vec{v}^{(N)} \quad (3.136)$$

El vector $\vec{v}^{(1)}$ es el vector de la mediana devuelto por este filtro. Este filtro permite reducir el ruido impulsivo, aunque puede considerar como ruido a puntos que realmente pertenecen a los bordes.

- Filtro de la media aritmética:

$$f_{VM}(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N) = \frac{1}{N} \sum_{i=1}^N \vec{v}_i \quad (3.137)$$

Este filtro reduce el ruido gaussiano, aunque puede detectar falsos puntos de borde. Este filtro es más eficiente que el filtro del vector mediana debido a que sólo requiere la operación de promediado.

- Filtro de la media α -ajustada (α -trimmed):

$$f_{\alpha-trim}(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N) = \frac{1}{N(1-2\alpha)} \sum_{i=1}^{N(1-2\alpha)} \vec{v}^{(i)} \quad (3.138)$$

donde $\alpha \in [0, 0.5)$. Cuando $\alpha = 0$, no se rechaza ningún vector, coincidiendo este filtro con el filtro de la media aritmética. Cuando $\alpha = 0.5$, se rechazan todos los vectores excepto el vector de la mediana y el filtro

se reduce al filtro del vector de la mediana. Para otros valores de α , este operador puede rechazar un $200\alpha\%$ de puntos de ruido impulsivo, siendo la salida el promedio de los vectores restantes. Por tanto, el filtro α -ajustada puede mejorar el rendimiento del detector en presencia de ruidos gaussiano e impulsivo.

- Filtro adaptativo del vecino más próximo (*adaptive nearest-neighbor filter*):

$$f_{adap}(\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N) = \sum_{i=1}^N w_i \vec{v}^{(i)} \quad (3.139)$$

donde la función w_i fue definida en 3.88 y asigna pesos altos a los vectores de menor rango y pesos bajos a los *outliers*. Este filtro también es efectivo ante ruidos gaussiano e impulsivo y posee una complejidad computacional similar a la del filtro α -ajustada, ya que ambos necesitan aplicar el método *R-ordering*. Sin embargo, la calidad de la detección de los bordes puede verse afectada por el hecho de que se utilizan vectores generados por el filtro en vez de los vectores originales.

Se pueden definir dos variantes de los detectores basados en la aplicación del detector *DV* y los filtros anteriores:

- Filtrar la imagen con alguno de los filtros reseñados, para lo cual se utilizaría ventanas de 3×3 puntos. A continuación se aplicaría el detector de bordes *DV*.

Este grupo de detectores es más eficiente que el anterior, ya que no se necesitan la división en subventanas y, además, la operación del filtrado sólo se efectúa una vez, mientras que en el caso anterior debía hacerse ocho veces (dos veces por cada una de las cuatro direcciones).

- Considerar sólo las direcciones vertical y horizontal, en vez de las cuatro direcciones. Esta variante puede ser aplicada a todos los detectores descritos anteriormente.

$$DV_{hv} = \max\{\|\vec{\nabla} \vec{f}_{0^\circ}\|, \|\vec{\nabla} \vec{f}_{90^\circ}\|\} \quad (3.140)$$

Esta simplificación está justificada por dos motivos:

- La visión humana es más sensible a los bordes horizontales y verticales que a los demás.
- Los vectores de diferencias horizontales y verticales pueden detectar la mayoría de los bordes diagonales, reduciendo su grosor al eliminar la redundancia de los detectores diagonales. Además, la complejidad computacional es ligeramente reducida.

3.4.3.3 Detectores de bordes en color basados en las derivadas de segundo orden

Los bordes de una imagen en color también pueden ser detectados en los puntos donde se producen los cruces por cero de las derivadas de segundo orden de la imagen. Dos métodos que han aplicado esta técnica son:

- Diferencias finitas de segundo orden (Gauch y Hsia 1992).
- Función de contraste multispectral (Cumani 1991; Cumani et al. 1991).

Diferencias finitas de segundo orden

Gauch y Hsia (1.992) definieron los bordes de una imagen como los ceros de la segunda derivada direccional de la imagen en la dirección del gradiente. Esto es equivalente a localizar los ceros de la derivada direccional de la magnitud del gradiente en la dirección del gradiente.

Para estimar el gradiente en un punto (x, y) de una imagen en color, se utilizan las siguientes diferencias finitas:

$$d\vec{f}_x(x, y) = (c_1(x+1, y) - c_1(x, y), c_2(x+1, y) - c_2(x, y), c_3(x+1, y) - c_3(x, y)) \quad (3.141)$$

$$d\vec{f}_y(x, y) = (c_1(x, y+1) - c_1(x, y), c_2(x, y) - c_2(x, y+1), c_3(x, y) - c_3(x, y+1)) \quad (3.142)$$

donde c_i ($i = 1, 2, 3$) representa cada una de las componentes del espacio de color⁷.

El gradiente es estimado como:

$$\vec{\nabla} \vec{f}(x, y) = (||d\vec{f}_x(x, y)||, ||d\vec{f}_y(x, y)||) \quad (3.143)$$

Las derivadas parciales del gradiente pueden ser estimadas usando las diferencias finitas de la magnitud del gradiente:

$$\frac{\partial \vec{\nabla} \vec{f}}{\partial x}(x, y) = ||\vec{\nabla} \vec{f}(x+1, y)|| - ||\vec{\nabla} \vec{f}(x, y)|| \quad (3.144)$$

$$\frac{\partial \vec{\nabla} \vec{f}}{\partial y}(x, y) = ||\vec{\nabla} \vec{f}(x, y+1)|| - ||\vec{\nabla} \vec{f}(x, y)|| \quad (3.145)$$

⁷Gauch y Hsia (1.992) aplicaron este método a imágenes definidas en los espacios de color *RGB*, *YIQ*, *HLS* y *Lab*.

Los bordes son identificados en los puntos donde se producen los cruces por cero de la siguiente derivada direccional:

$$D(x, y) = \left| \frac{\partial \vec{\nabla} \vec{f}}{\partial x}(x, y) \right| \cdot \frac{\partial \vec{\nabla} \vec{f}}{\partial x}(x, y) + \left| \frac{\partial \vec{\nabla} \vec{f}}{\partial y}(x, y) \right| \cdot \frac{\partial \vec{\nabla} \vec{f}}{\partial y}(x, y) \quad (3.146)$$

Función de contraste multiespectral

Una técnica más compleja para identificar los bordes mediante las derivadas de segundo orden está basada en “la función de contraste multiespectral” (Alshatti y Lambert 1993; Cumani 1991; Cumani et al. 1991; Koschan 1995). Cumani definió el valor de la función de contraste multiespectral de una imagen \vec{f} en un punto $P = (x, y)$ en la dirección del vector unitario $\vec{n} = (n_1, n_2)$ como

$$S(P, \vec{n}) = E(x, y)n_1^2 + F(x, y)n_1n_2 + G(x, y)n_2^2 \quad (3.147)$$

donde⁸

$$\begin{aligned} E(x, y) &= \frac{\partial \vec{f}(x, y)}{\partial x} * \frac{\partial \vec{f}(x, y)}{\partial x} \\ &= \frac{\partial R}{\partial x}(x, y) \frac{\partial R}{\partial x}(x, y) + \frac{\partial G}{\partial x}(x, y) \frac{\partial G}{\partial x}(x, y) + \frac{\partial B}{\partial x}(x, y) \frac{\partial B}{\partial x}(x, y) \end{aligned} \quad (3.148)$$

$$\begin{aligned} F(x, y) &= \frac{\partial \vec{f}(x, y)}{\partial x} * \frac{\partial \vec{f}(x, y)}{\partial y} \\ &= \frac{\partial R}{\partial x}(x, y) \frac{\partial R}{\partial y}(x, y) + \frac{\partial G}{\partial x}(x, y) \frac{\partial G}{\partial y}(x, y) + \frac{\partial B}{\partial x}(x, y) \frac{\partial B}{\partial y}(x, y) \end{aligned} \quad (3.149)$$

$$\begin{aligned} G(x, y) &= \frac{\partial \vec{f}(x, y)}{\partial y} * \frac{\partial \vec{f}(x, y)}{\partial y} \\ &= \frac{\partial R}{\partial y}(x, y) \frac{\partial R}{\partial y}(x, y) + \frac{\partial G}{\partial y}(x, y) \frac{\partial G}{\partial y}(x, y) + \frac{\partial B}{\partial y}(x, y) \frac{\partial B}{\partial y}(x, y) \end{aligned} \quad (3.150)$$

⁸Los valores de $E(x, y)$, $F(x, y)$ y $G(x, y)$ coinciden, respectivamente, con los valores $g_{xx}(x, y)$, $g_{yy}(x, y)$ y $g_{xy}(x, y)$ indicados en las ecuaciones 3.93, 3.94 y 3.95. Sin embargo, se ha preferido respetar la notación utilizada por Cumani.

Los valores extremos de $S(P, \vec{n})$ coinciden con los autovalores de la matriz

$$\begin{bmatrix} E(x, y) & F(x, y) \\ F(x, y) & G(x, y) \end{bmatrix} \quad (3.151)$$

Los autovalores son:

$$\lambda_{\pm}(x, y) = \frac{1}{2}(E(x, y) + G(x, y) \pm \sqrt{(E(x, y) - G(x, y))^2 + 4F^2(x, y)}) \quad (3.152)$$

y los autovectores correspondientes \vec{n}_+ y \vec{n}_- son definidos como

$$\vec{n}_{\pm} = (\cos(\theta_{\pm}), \sin(\theta_{\pm})) \quad (3.153)$$

donde

$$\theta_+ = \begin{cases} \frac{\pi}{4} & \text{Si } (E(x, y) - G(x, y)) = 0 \text{ y } F(x, y) > 0 \\ -\frac{\pi}{4} & \text{Si } (E(x, y) - G(x, y)) = 0 \text{ y } F(x, y) < 0 \\ \text{Indefinido} & \text{Si } E(x, y) = F(x, y) = G(x, y) = 0 \\ \frac{1}{2} \arctan \left(\frac{2F(x, y)}{(E(x, y) - G(x, y))} \right) + k\pi & \text{En otro caso} \end{cases} \quad (3.154)$$

y

$$\theta_- = \theta_+ \pm \frac{\pi}{2} \quad (3.155)$$

Los valores de 3.152 y 3.153 representan una extensión multispectral del concepto de gradiente y por ello parece natural considerar los cruces por cero de la derivada de $\lambda_+(x, y)$ en la dirección de \vec{n}_+ como puntos candidatos de la detección de bordes. Dicha derivada direccional es expresada por

$$\begin{aligned} D_S(P, \vec{n}) &= \nabla \lambda_+(x, y) \cdot \vec{n}_+ \\ &= E_x(x, y)n_1^3 \\ &\quad + (2F_x(x, y) + E_y(x, y))n_1^2n_2 \\ &\quad + (G_x(x, y) + 2F_y(x, y))n_1n_2^2 \\ &\quad + G_y(x, y)n_2^3 \end{aligned} \quad (3.156)$$

donde el subíndice “+” ha sido suprimido de las componentes de \vec{n} por simplicidad.

Los puntos de los bordes son localizados en los cruces por cero de $D_S(P, \vec{n})$, considerando el signo de D_S a lo largo de la curva tangente a n_+ en P .

Para calcular el valor de $D_S(P, \vec{n})$, se necesitan las derivadas parciales E_x , E_y , F_x , F_y , G_x y G_y . Puesto que calcular y almacenar los valores de E , F y G y después computar sus derivadas parciales requiere una gran cantidad de tiempo

de ejecución, Koschan (1.995) propuso un método más eficiente para calcular directamente dichas derivadas parciales:

$$E_x = 2 \left(\frac{\partial R}{\partial x} \frac{\partial^2 R}{\partial x^2} + \frac{\partial G}{\partial x} \frac{\partial^2 G}{\partial x^2} + \frac{\partial B}{\partial x} \frac{\partial^2 B}{\partial x^2} \right) \quad (3.157)$$

$$E_y = 2 \left(\frac{\partial R}{\partial x} \frac{\partial^2 R}{\partial x \partial y} + \frac{\partial G}{\partial x} \frac{\partial^2 G}{\partial x \partial y} + \frac{\partial B}{\partial x} \frac{\partial^2 B}{\partial x \partial y} \right) \quad (3.158)$$

$$F_x = \left(\frac{\partial R}{\partial x} \frac{\partial^2 R}{\partial x \partial y} + \frac{\partial R}{\partial y} \frac{\partial^2 R}{\partial x^2} \right) + \left(\frac{\partial G}{\partial x} \frac{\partial^2 G}{\partial x \partial y} + \frac{\partial G}{\partial y} \frac{\partial^2 G}{\partial x^2} \right) + \left(\frac{\partial B}{\partial x} \frac{\partial^2 B}{\partial x \partial y} + \frac{\partial B}{\partial y} \frac{\partial^2 B}{\partial x^2} \right) \quad (3.159)$$

$$F_y = \left(\frac{\partial R}{\partial y} \frac{\partial^2 R}{\partial x \partial y} + \frac{\partial R}{\partial x} \frac{\partial^2 R}{\partial y^2} \right) + \left(\frac{\partial G}{\partial y} \frac{\partial^2 G}{\partial x \partial y} + \frac{\partial G}{\partial x} \frac{\partial^2 G}{\partial y^2} \right) + \left(\frac{\partial B}{\partial y} \frac{\partial^2 B}{\partial x \partial y} + \frac{\partial B}{\partial x} \frac{\partial^2 B}{\partial y^2} \right) \quad (3.160)$$

$$G_x = 2 \left(\frac{\partial R}{\partial y} \frac{\partial^2 R}{\partial x \partial y} + \frac{\partial G}{\partial y} \frac{\partial^2 G}{\partial x \partial y} + \frac{\partial B}{\partial y} \frac{\partial^2 B}{\partial x \partial y} \right) \quad (3.161)$$

$$G_y = 2 \left(\frac{\partial R}{\partial y} \frac{\partial^2 R}{\partial y^2} + \frac{\partial G}{\partial y} \frac{\partial^2 G}{\partial y^2} + \frac{\partial B}{\partial y} \frac{\partial^2 B}{\partial y^2} \right) \quad (3.162)$$

Para el cálculo de las derivadas parciales, Cumani y Alshatti propusieron el uso de máscaras de convolución de tamaño 3×3 , mientras que Koschan (1.995) sugirió la utilización de máscaras gaussianas de mayor tamaño.

Independientemente de cómo hayan sido calculadas las derivadas parciales, este método presenta una dificultad en la localización de los puntos de los bordes debida a la ambigüedad existente en la determinación de la dirección del gradiente. Para resolver este problema se han propuesto dos alternativas:

- Aplicar una técnica de interpolación bilineal, denominada “*subpixel resolution*”, para determinar la correcta ubicación de los puntos de borde (Cumani 1991).
- Estimar directamente el valor del autovector \vec{n}_+ (Alshatti y Lambert 1993).

Los operadores basados en la segunda derivada son muy sensibles a variaciones de textura y al ruido impulsivo, pero generan bordes más delgados (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999).

3.4.3.4 Detectores de bordes en color que utilizan la entropía

La entropía también ha sido utilizada para detectar bordes en imágenes en color (Fotinos et al. 1999; Shiozaki 1986). Si la cromaticidad del color en una región local es uniforme entonces los operadores basados en la entropía generan un valor grande; por el contrario, si los cambios de cromaticidad del color son drásticos entonces el valor generado es pequeño.

Se define la entropía H en una ventana de tamaño 3×3 centrada sobre un punto de la imagen $f(x_0, y_0) = (r_0, g_0, b_0)$ como

$$H = q_R H_R + q_G H_G + q_B H_B \quad (3.163)$$

donde

$$q_R = \frac{r_0}{r_0 + g_0 + b_0} \quad (3.164)$$

$$q_G = \frac{g_0}{r_0 + g_0 + b_0} \quad (3.165)$$

$$q_B = \frac{b_0}{r_0 + g_0 + b_0} \quad (3.166)$$

y H_R , H_G y H_B denotan las entropías en las direcciones R , G y B , respectivamente. El valor H_X ($X \in \{R, G, B\}$) se define como

$$H_X = -\frac{\sum_{i=1}^N p_{X_i} \log(p_{X_i})}{\log(N)} \quad (3.167)$$

donde

$$p_{X_i} = \frac{X_i}{\sum_{j=1}^N X_j} \quad (3.168)$$

Los bordes pueden ser obtenidos detectando los puntos que tengan una menor entropía (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999). Puesto que la presencia del ruido puede perturbar la cromaticidad local de una imagen, los operadores basados en la entropía son sensibles al ruido (Trahanias y Venetsanopoulos 1993).

3.4.3.5 Detectores de bordes en color basados en técnicas de *clustering*

Las técnicas de *clustering* o de agrupamiento de clases también han sido utilizadas para diseñar algoritmos de detección de bordes en color:

- Huntsberger y Descalzi (1.985) consideraron que los puntos de una imagen en color podrían ser caracterizados mediante sus valores de pertenencia difusa (*fuzzy*) a grupos (*clusters*) de un espacio de color vectorial. Estos valores de pertenencia permiten definir un borde de color que no depende de un umbral. Un borde de color es definido como el cruce por cero de las diferencias entre los valores de pertenencia en cada punto.
- Tao y Huang (1.997) utilizaron un método de *clustering* para obtener n regiones o *cluster* en cada imagen en color. Cada una de estas regiones o *clusters* puede ser caracterizada por medio del vector asociado al centro de la región o *cluster*. Si \vec{v}_1 y \vec{v}_2 son los vectores asociados a dos regiones R_1 y R_2 , respectivamente, entonces el vector $\vec{v} = \vec{v}_1 - \vec{v}_2$ tiene una dirección similar a la del vector gradiente de un punto que esté situado en el borde que separa a ambas regiones. Si la imagen tiene n *clusters*, entonces hay $\frac{n(n-1)}{2}$ posibles vectores. Para cada punto de la imagen, su vector gradiente es proyectado sobre todos estos $\frac{n(n-1)}{2}$ vectores y el valor máximo es utilizado como magnitud del borde en dicho punto.

Los inconvenientes de este tipo de métodos son:

- El número de *clusters* de una imagen suele ser desconocido y puede variar de una imagen a otra.
- Si el número de *clusters* es elevado, la complejidad computacional aumenta considerablemente.

Los dos métodos indicados superaron estos inconvenientes utilizando un número fijo de *clusters*: 4 en el algoritmo de Huntsberger y Descalzi y 6 en el de Tao y Huang. Sin embargo, esta simplificación puede no ser válida para detectar bordes en todas las imágenes de color.

3.4.3.6 Detector de bordes en color basado en histogramas

Los histogramas también han sido utilizados para detectar bordes en imágenes en color. Pietikäinen y Harwood (1.986) propusieron un detector basado en el uso de histogramas: las diferencias de color entre cada punto y sus vecinos son utilizadas para obtener un histograma de diferencias tridimensional, a partir del cual se obtiene un histograma unidimensional de frecuencias acumuladas. Este histograma es convertido a porcentajes de distribución de las frecuencias acumuladas.

El valor de borde de color en un punto se define como el máximo porcentaje de las frecuencias acumuladas de las diferencias de color entre dicho punto y sus ocho vecinos. Finalmente, los bordes son identificados mediante umbralización.

3.4.3.7 Detector de bordes en color basado en la teoría bayesiana

Se ha propuesto un método de detección de bordes en color que utiliza un criterio derivado de la teoría bayesiana (Djurić y Fwu 1997). Este detector consta de dos etapas:

1. Detección de los bordes potenciales: se aplican las máscaras de la figura 3.28 a cada una de las componentes de color de la imagen para determinar qué puntos son candidatos a pertenecer al mapa de bordes de la imagen.

	0.1	- 0.1			0.1	
0.1	0.7	- 0.7	- 0.1		0.1	0.7
	0.1	- 0.1			- 0.1	- 0.7
					-0.1	
(a)				(b)		

Figura 3.28: Máscaras para detectar bordes (a) verticales y (b) horizontales.

2. Utilización de los bordes candidatos y del principio de la “máxima probabilidad a posteriori” (MAP, *maximum a posteriori probability*) para estimar el número de bordes y su localización.

3.4.3.8 Detector de bordes en color basado en polinomios ortogonales

Se ha propuesto un algoritmo de detección de bordes en imágenes definidas en el espacio de color *RGB* que utiliza un conjunto de polinomios ortogonales (Krishnamoorthi y Bhattacharyya 1997).

El algoritmo tiene en cuenta la correlación espacial dentro de cada plano de color y la interacción entre los diferentes planos. A partir de un conjunto de polinomios ortogonales, se define un conjunto de operadores de diferencia. Estos operadores son empleados para representar una región de una imagen en color como una combinación lineal de las respuestas de los operadores provocadas por los bordes en color y el ruido. Las respuestas de los bordes y las del ruido son separadas mediante un análisis estadístico. El detector de bordes en color está basado en la maximización de la proporción entre la señal y el ruido (*signal to noise ratio*, *SNR*).

Capítulo 4

Estudios comparativos de algoritmos de detección de bordes

4.1 Introducción

La evaluación de la calidad de cualquier tipo de algoritmo tiene una gran importancia, independientemente del ámbito científico en el que se sitúe dicho algoritmo. La evaluación del rendimiento de un algoritmo es una cuestión esencial para la evolución del procesamiento digital de imágenes (Haralick 1994).

Las principales características que debe poseer un detector de bordes son (Plataniotis y Venetsanopoulos 2000):

- Identificar los bordes con exactitud.
- Ser robusto en presencia de ruido.
- Ser computacionalmente eficiente.

Para poder valorar la calidad de un detector es necesario utilizar un método de evaluación justo y efectivo. Aunque se han propuesto numerosos métodos de evaluación de los detectores de bordes, no existe ninguno que sea considerado como el método estándar.

Los métodos de evaluación de algoritmos de procesamiento digital de imágenes pueden ser clasificados como objetivos o subjetivos. La evaluación objetiva permite valorar cuantitativamente el rendimiento de los algoritmos. Sin embargo, en la mayoría de las aplicaciones, la evaluación humana es la que establece la decisión final (Clinque et al. 1994; Plataniotis y Venetsanopoulos 2000).

En este capítulo se van a describir las características generales de los diferentes estudios comparativos de los algoritmos de detección de bordes en imágenes monocromáticas y en color.

4.2 Características generales de los estudios comparativos

Se han realizado numerosos estudios comparativos que evalúan la calidad de los algoritmos de detección de bordes en imágenes monocromáticas (tablas 4.2 - 4.5), mientras que son muy pocos los que han evaluado los algoritmos de detección de bordes en color (Allen y Huntsberger 1989; Gauch y Hsia 1992; Koschan 1995; Novak y Shafer 1987; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999) (tablas 4.6 - 4.8).

Además de estos estudios, existen algunas recopilaciones de detectores de bordes en imágenes monocromáticas (Davis 1975; Fu y Mui 1981; Pal y Pal 1993; Torre y Poggio 1986; Ziou y Tabbone 1998) o en color (Skarbek y Koschan 1994). Sin embargo, estos trabajos no evalúan el rendimiento de los algoritmos, aunque sí analizan los fundamentos teóricos en los que se basan.

Los estudios comparativos poseen ciertas características generales que pueden ser descritas respondiendo a las siguientes cuestiones:

- Propósito de la evaluación: ¿cuál es el objetivo principal de la evaluación?
- Tipo de evaluación: ¿qué metodología ha sido utilizada?
- Tipos de imágenes: ¿qué características tienen las imágenes utilizadas?
- Tipos de bordes: ¿Qué tipos de bordes son utilizados? ¿Cuál es su forma? ¿Cómo han sido generados?
- Detectores de bordes evaluados: ¿cuáles son los algoritmos comparados? ¿Pertencen los algoritmos a clases diferentes o son todos del mismo tipo?

Estas cuestiones son desarrolladas en las siguientes subsecciones.

4.2.1 Propósito de la evaluación

Los estudios comparativos de los detectores de bordes han sido realizados por diferentes motivos. Algunos de los motivos más frecuentes e importantes son:

- Medir la capacidad de detección y localización de los detectores de bordes. Este es el principal objetivo de la mayoría de los estudios comparativos.
- Analizar la influencia de una o más características (ruido, contraste del borde, orientación del borde, etc.) sobre los detectores de bordes (Fram y Deustsch 1975; Kanungo et al. 1995; Lyvers y Mitchell 1988).
- Evaluar la buena forma de los bordes detectados, especialmente la continuidad y la delgadez (Kitchen y Rosenfeld 1981; Zhu 1996)
- Comprobar si los detectores poseen algunas propiedades matemáticas (Owens y Venkatesh 1989) o estadísticas (Panda y Dubitzki 1979).
- Comparar diferentes detectores de bordes entre sí con respecto a una o varias medidas (Allen y Huntsberger 1989; Bernsen 1991; Heath et al. 1996; Heath et al. 1997; Heath et al. 1998; Koschan 1995; Zhu et al. 1999).
- Presentar una nueva metodología para la evaluación de los detectores de bordes (Cho et al. 1997; Dougherty y Bowyer 1998; Heath et al. 1996; Heath et al. 1997; Heath et al. 1998; Spreeuwers y van der Heijden 1992; Strickland y Cheng 1993; Venkatesh y Kitchen 1992).
- Comparar un nuevo algoritmo de detección de bordes con otros algoritmos ya existentes (Haralick 1984).

Estos motivos no son excluyentes entre sí, sino todo lo contrario. En realidad, la mayoría de los estudios comparativos han sido realizados para satisfacer los cinco primeros motivos.

4.2.2 Tipo de evaluación

Los estudios comparativos han utilizado diferentes metodologías que pueden ser clasificadas como:

- Teóricas: analizan los fundamentos teóricos de los algoritmos.
- Experimentales: aplican los algoritmos a diversas imágenes y evalúan los resultados obtenidos.

Evaluación teórica

Los estudios comparativos que realizan una evaluación teórica analizan matemáticamente los algoritmos, pero no los aplican computacionalmente a ninguna imagen. La entrada de cada algoritmo es caracterizada matemáticamente y su rendimiento es determinado analíticamente (Abdou y Pratt 1979; Lyvers y Mitchell

1988; Owens y Venkatesh 1989; Panda y Dubitzki 1979; Ramesh y Haralick 1992a; Ramesh y Haralick 1992b; Ziou y Fabre 1994; Ziou et al. 1994).

Las principales dificultades de este tipo de evaluación son:

- Ha de caracterizar correctamente, mediante modelos matemáticos, las imágenes de entrada y el ruido.
- Ha de superar la complejidad matemática de los detectores de bordes más modernos.

Evaluación experimental

Los estudios comparativos que realizan una evaluación experimental aplican los algoritmos a una o varias imágenes reales o artificiales y utilizan las imágenes resultantes para medir su rendimiento.

Los estudios comparativos basados en la evaluación experimental pueden ser clasificados de dos maneras diferentes, dependiendo del criterio que se utilice:

- Uso de imágenes de referencia¹:
 - Evaluación con imágenes de referencia.
 - Evaluación sin imágenes de referencia.
- Método de evaluación:
 - Objetiva o cuantitativa.
 - Subjetiva o cualitativa.

Un estudio comparativo puede ser objetivo o subjetivo independientemente de que utilice o no imágenes de referencia.

Los estudios comparativos que utilizan una imagen de referencia miden las diferencias entre esta imagen y la imagen de salida producida por el detector. Estos estudios se denominan “estudios comparativos de discrepancia”. La mayoría de los estudios comparativos de detección de bordes pueden encuadrarse en esta clase (Abdou 1978; Abdou y Pratt 1979; Bernsen 1991; Bryant y Bouldin 1979; Fram y Deustsch 1975; Haralick 1984; Jiang et al. 1995; Kanungo et al. 1995; Lyvers y Mitchell 1988; Owens y Venkatesh 1989; Peli y Malah 1982; Salotti et al. 1996; Shin et al. 1998a; Spreeuwers y van der Heijden 1992; Strickland y Cheng 1993; Venkatesh y Kitchen 1992).

¹La literatura científica denomina a la información contenida en estas imágenes *ground truth*.

La mayoría de las imágenes de referencia utilizadas han sido artificiales o sintéticas, debido a que en dichas imágenes se puede identificar fácilmente los mapas de bordes. No obstante, también se han realizado estudios en los que se ha hecho uso de imágenes de referencia reales en los que el mapa de bordes podía ser definido manualmente (Dougherty y Bowyer 1998; Salotti et al. 1996; Shin et al. 1998a; Shin et al. 1998b).

La principal dificultad de este tipo de evaluación basada en la discrepancia entre imágenes radica en la correcta selección de las imágenes de referencia, ya que no existe ningún conjunto finito de imágenes - reales o artificiales - que tenga la complejidad suficiente para poder representar a todas las escenas reales posibles.

Los estudios comparativos sin imágenes de referencia miden “la bondad” de los algoritmos, ya que evalúan la calidad de sus resultados, como pueden ser el grado de coincidencia entre diferentes detectores (Bryant y Bouldin 1979), la forma, continuidad y delgadez de los bordes (Kitchen y Rosenfeld 1981; Zhu 1996), la verosimilitud de que un borde detectado sea un borde verdadero (Cho et al. 1997; Ramesh y Haralick 1992a; Ramesh y Haralick 1992b), la identificación de algún patrón (Bernsen 1991), etc.

La principal dificultad de la evaluación experimental sin imágenes de referencia es que no puede medir el desplazamiento de los bordes respecto de sus verdaderas posiciones.

Los estudios comparativos son objetivos si aplican una medida cuantitativa a la imagen de salida para evaluar la calidad del detector. La mayoría de los estudios experimentales realizan una comparación objetiva de los detectores de bordes.

Los métodos cuantitativos u objetivos que utilizan imágenes de referencia pueden ser agrupados en dos clases:

- Métodos probabilísticos: calculan las probabilidades de detección correcta y de falsa detección.
- Métodos basados en distancias: miden la desviación de los bordes detectados respecto de los bordes verdaderos.

También se ha propuesto estudios cuantitativos que combinan de forma ponderada varias medidas de error (Spreeuwens y van der Heijden 1992; Strickland y Cheng 1993; Venkatesh y Kitchen 1992).

Los estudios subjetivos evalúan la calidad de cada detector en función de las opiniones realizadas por una o varias personas acerca de la calidad de las imágenes generadas (Fram y Deustsch 1975; Koschan 1995; Peli y Malah 1982). Algunas veces estas valoraciones son analizadas estadísticamente (Heath et al. 1997).

Casi todos los estudios experimentales incluyen una comparación visual y subjetiva para contrastar los resultados obtenidos con las pruebas objetivas.

Algunos estudios comparativos incluyen diferentes tipos de evaluaciones, tanto teóricas como experimentales (Abdou 1978; Owens y Venkatesh 1989; Panda y Dubitzki 1979), cuantitativas y cualitativas (Bernsen 1991; Haralick 1984; Zhu et al. 1999), con imágenes de referencia o sin ellas (Bryant y Bouldin 1979; Heath et al. 1996; Heath et al. 1997; Heath et al. 1998).

4.2.3 Tipos de imágenes utilizados y tipos de bordes que contienen

Las imágenes utilizadas en las pruebas experimentales han sido de dos tipos:

- Imágenes artificiales o sintéticas.
- Imágenes reales.

Hay que hacer constar que son muy pocos los estudios comparativos que indican la definición espacial (número de puntos) y la resolución digital (número de niveles de gris)² de las imágenes utilizadas, lo cual no permite enjuiciar completamente el método de evaluación.

Bordes contenidos en imágenes artificiales

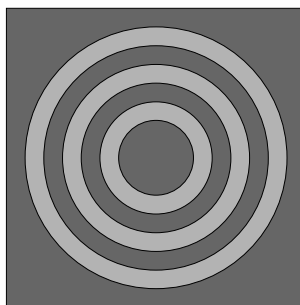
Las imágenes artificiales o sintéticas son creadas específicamente para evaluar la calidad de los algoritmos seleccionados. Los bordes que aparecen en estas imágenes son generalmente de los tipos “escalón” o “rampa”. Estos bordes son generados por figuras geométricas (círculos, rectángulos, cuadrados, aros, escuadras, etc.) que dan lugar a líneas rectas o curvas (Abdou y Pratt 1979; Bernsen 1991; Fram y Deustsch 1975; Haralick 1984; Heath et al. 1996; Kanungo et al. 1995; Kitchen y Rosenfeld 1981; Lyvers y Mitchell 1988; Peli y Malah 1982; Owens y Venkatesh 1989; Venkatesh y Kitchen 1992; Zhu et al. 1999). También se han generado bordes utilizando otros modelos geométricos, como los mosaicos de Voronoi (Spreeuwens y van der Heijden 1992), o modelos pseudoaleatorios (Panda y Dubitzki 1979; Ramesh y Haralick 1992a; Ziou y Tabbone 1998).

En el caso de las imágenes artificiales en color, los bordes también son generados por variaciones en alguna o en varias de las componentes del espacio de color (Allen y Huntsberger 1989; Koschan 1995; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999).

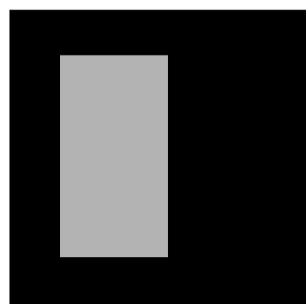
²En las imágenes en color, se refiere al número de niveles de cada banda del espacio de color.



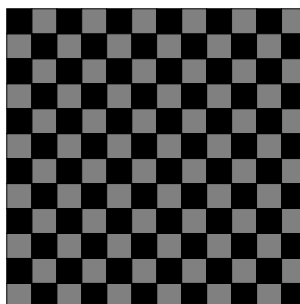
(a)



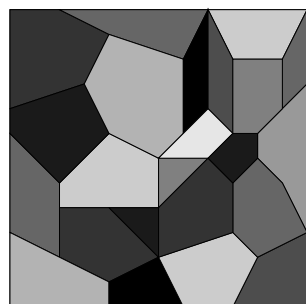
(b)



(c)



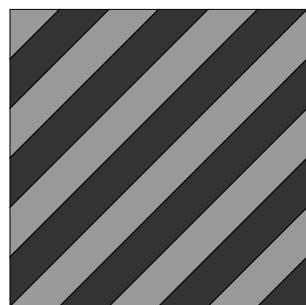
(d)



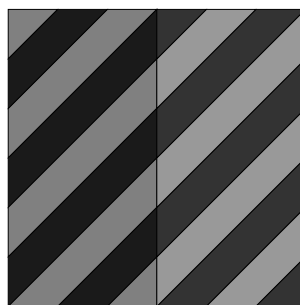
(e)



(f)



(g)



(h)

Figura 4.1: Tipos de imágenes utilizadas en los estudios de (a y b) Kitchen y Rosenfeld (1.981), (c y d) Owens y Venkatesh (1.989), (e) Spreeuwers y van der Heijden (1.992), (f) Palmer *et al.* (1.996) y (g y h) Kanungo *et al.* (1.995).

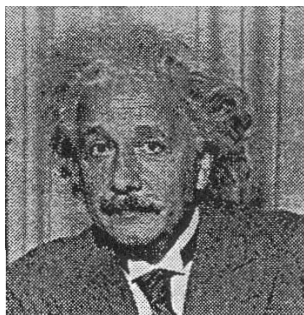
Una vez generadas las imágenes artificiales, se les suele añadir diferentes niveles de ruido. Generalmente, el ruido añadido es de tipo gaussiano, aunque algunas veces se ha añadido ruido impulsivo o de “sal y pimienta” (Peli y Malah 1982; Zhu et al. 1999).

Bordes contenidos en imágenes reales

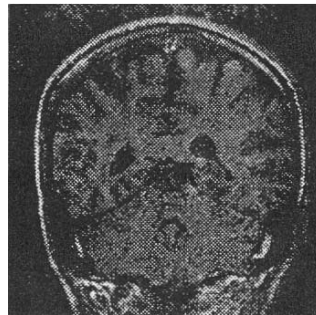
El uso de las imágenes reales permite comprobar el comportamiento del detector ante situaciones en las que posteriormente será utilizado.

Dougherty y Bowyer (1.998) clasificaron las imágenes reales utilizadas en los estudios comparativos en cuatro categorías:

- Imágenes de objetos (Abdou y Pratt 1979; Bernsen 1991; Heath et al. 1998).
- Imágenes aéreas (Abdou y Pratt 1979; Bryant y Bouldin 1979; Fram y Deustsch 1975; Haralick 1984).
- Imágenes médicas del cuerpo humano (Dougherty y Bowyer 1998).
- Imágenes de rostros humanos (Abdou y Pratt 1979; Owens y Venkatesh 1989; Ramesh y Haralick 1992a; Salotti et al. 1996).



(a)



(b)

Figura 4.2: Imágenes utilizadas por Ramesh y Haralick (1.992): (a) Einstein y (b) un cerebro.

No obstante, éstos no son los únicos tipos de imágenes utilizados, ya que, por ejemplo, también se han utilizado imágenes de texturas como las del libro de Brodatz (1.966) (Panda y Dubitzki 1979).

Los bordes contenidos en las imágenes reales son más complejos que los generados en las imágenes artificiales, aunque se pueden buscar modelos reales similares a los tipos de bordes³ más usuales (“escalón” o “rampa”).

³Véase la sección 3.2.

La mayoría de los estudios comparativos utilizan las imágenes reales para realizar una valoración cualitativa de los detectores de bordes. Sin embargo, algunos estudios también han utilizado las imágenes reales como imágenes de referencia para efectuar mediciones cuantitativas. El uso de imágenes reales como imágenes de referencia requiere la determinación de los mapas de bordes, lo cual constituye un problema de difícil solución. No obstante, algunos investigadores han propuesto que se definan los mapas de bordes de forma manual (Bryant y Bouldin 1979; Dougherty y Bowyer 1998; Salotti et al. 1996; Shin et al. 1998a).

4.2.4 Tipos de detectores evaluados

La mayoría de los estudios comparativos de detectores de bordes han evaluado algoritmos de “realce” y “umbralización”, como, por ejemplo, los detectores clásicos (Roberts, Sobel, Prewitt, etc.). Sin embargo, no ha sido éste el único tipo de detector evaluado, ya que también se han evaluado los detectores basados en el ajuste paramétrico, o en los cruces por cero de la segunda derivada, o en la supresión de puntos que no eran máximos locales en la dirección del gradiente, etc.

Algunos estudios se han centrado en el análisis de algoritmos pertenecientes a una misma clase, es decir, basados en los mismos fundamentos matemáticos (Abdou y Pratt 1979; Cho et al. 1997; Kitchen y Rosenfeld 1981; Peli y Malah 1982; Shin et al. 1998a; Venkatesh y Kitchen 1992; Ziou y Fabre 1994), mientras que otros han comparado algoritmos de detección de bordes de clases diferentes (Bernsen 1991; Dougherty y Bowyer 1998; Fram y Deustsch 1975; Fries y Modestino 1977; Haralick 1984; Heath et al. 1996; Heath et al. 1997; Heath et al. 1998; Kitchen y Rosenfeld 1981; Lyvers y Mitchell 1988; Owens y Venkatesh 1989; Ramesh y Haralick 1992a; Spreeuwers y van der Heijden 1992; Strickland y Cheng 1993; Zhu et al. 1999).

Además, dos de los estudios comparativos han sido aplicados sólo a algoritmos de detección de líneas (Kanungo et al. 1995; Palmer et al. 1996).

4.3 Evaluación de algoritmos de detección de bordes en imágenes monocromáticas

Son muy numerosos los estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas. Las tablas 4.2-4.5 resumen las características básicas de muchos de ellos.

A continuación, sólo se van a describir los siguientes estudios comparativos:

- Calificaciones relativa y absoluta de los detectores de bordes (Bryant y Bouldin 1979).
- Evaluación realizada por Abdou y Pratt (1.979).
- Uso de las componentes necesarias de un espacio vectorial de errores para la evaluación de los detectores de bordes (Venkatesh y Kitchen 1992).

Estos métodos han sido elegidos porque que serán utilizados posteriormente en la evaluación de los algoritmos de detección de bordes en imágenes en color⁴.

Se va a realizar una descripción general de cada estudio comparativo, que va a incluir la cita de los detectores de bordes evaluados, las características de las imágenes usadas en las pruebas experimentales, la explicación de la metodología de evaluación utilizada o propuesta y la reseña de las principales conclusiones obtenidas por los autores.

4.3.1 Calificaciones relativa y absoluta de los detectores de bordes

Descripción

Bryant y Bouldin (1.979) propusieron dos métodos para evaluar cuantitativamente los detectores de bordes:

- Calificación relativa.
- Calificación absoluta.

Los autores indicaron que las medidas propuestas no debían considerarse como medidas exclusivas para determinar la calidad de un detector de bordes, sino que deberían ser empleadas para evaluar los detectores de bordes en un entorno específico. Estas medidas pueden ser utilizadas para seleccionar un detector que desarrolle una tarea particular, para diseñar nuevos detectores o para identificar los bordes principales en una imagen (Bryant y Bouldin 1979).

Detectores de bordes comparados

Los operadores comparados por Bryant y Bouldin fueron:

- Operador del gradiente (González y Wints 1978)

⁴Véase el capítulo 8.

- Operador de Roberts (1.965):
- Operador de rango o de máxima diferencia (Davies 1978):
- Operador de Sobel (Sobel 1978; Pratt 1978).
- Operador de Frei y Chen (1.997)

Para poder comparar los detectores de bordes, se hizo uso de una fase final de umbralización. El método elegido por Bryant y Bouldin fue el de umbralización porcentual mediante “ensayo y error”. Los porcentajes utilizados para todos los detectores fueron del 10% y 15%.

Características de las imágenes utilizadas

Los detectores de bordes son aplicados a dos imágenes de fotografías aéreas realizadas sobre un área rural. Las imágenes contienen bordes largos e irregulares.

Para realizar las pruebas de calificación absoluta, se definió de forma manual un mapa de bordes en cada una de las imágenes reales utilizadas.

Hay que hacer constar que los autores no indican la definición espacial (número de puntos) ni la resolución digital (número de niveles de gris) de las imágenes.

Medidas de evaluación propuestas

Bryant y Bouldin propusieron dos medidas de evaluación cuantitativa de los detectores de bordes:

- Calificación relativa: mide la coincidencia del resultado de cada detector con los obtenidos por los demás.

Para poder comparar los detectores de bordes, este método utiliza una imagen de “consenso” obtenida de la siguiente manera: un punto no pertenece a ella si hay un cierto número de operadores que no lo consideran como punto de borde, es decir, no lo han detectado; en caso contrario, el punto pertenece a la imagen de consenso.

El número mínimo de detectores requeridos para excluir un punto de la imagen de consenso puede variar. Los autores utilizaron dos criterios, uno en el que un punto era excluido de la imagen de consenso si no era detectado por al menos dos operadores y otro en el que se exigía que el punto no fuera localizado por al menos tres operadores. La utilización de un criterio u otro influye bastante en la composición de la imagen de consenso.

Una vez obtenida la imagen de consenso, se calculan los siguientes valores para cada detector:

- Porcentaje de puntos de borde de la imagen de consenso que ha detectado como puntos de borde.
- Porcentaje de puntos que no son de borde de la imagen de consenso que no ha detectado como puntos de borde.
- Porcentaje total de puntos (de borde y no borde) de la imagen de consenso que el detector ha clasificado correctamente.

El último valor es el que da la medida relativa del detector de bordes. Los otros dos valores muestran, respectivamente, el comportamiento diferenciado del detector ante puntos que pertenecen a los bordes y ante puntos que no pertenecen.

- Calificación absoluta: se compara la imagen generada por el detector de bordes con una imagen “correcta” o “clave” (*key image*) construida “manualmente”, que puede ser considerada como la salida de un detector ideal. El grado de coincidencia entre ambas imágenes mide la calidad del detector.

Un método usual para medir la similitud entre la salida real y la correcta es la correlación R definida como:

$$R(x, y) = \frac{\sum_{i=1}^I \sum_{j=1}^J K(i, j) O(x + i - 1, y + j - 1)}{\sqrt{\sum_{i=1}^I \sum_{j=1}^J K^2(i, j)} \sqrt{\sum_{i=1}^I \sum_{j=1}^J O^2(x + i - 1, y + j - 1)}} \quad (4.1)$$

donde K es la imagen correcta de tamaño $I \times J$ y O es la imagen generada por el detector, cuyo tamaño es $N \times M$.

Los autores indican que la correlación no es el único método que permite calcular la calidad absoluta. Se puede utilizar cualquier otro método que tenga en cuenta la relación entre la calidad y el borde ideal.

4.3.2 Evaluación realizada por Abdou y Pratt

Descripción

Abdou y Pratt (1.979) presentaron unas técnicas cuantitativas para el diseño y la evaluación de algoritmos de detección basados en el realce y la umbralización de los bordes.

Las técnicas de diseño que presentaron estaban basadas en la teoría estadística de la detección y en procedimientos de clasificación determinísticos de reconocimiento de patrones.

Los métodos de evaluación utilizados incluían:

- La medida determinística de la magnitud del gradiente del borde.
- La comparación de las probabilidades de correcta y falsa detección del borde.
- La medida de Pratt⁵.

Completaron su estudio con una evaluación subjetiva del funcionamiento de los detectores de bordes.

Detectores de bordes comparados

Los autores se centraron en el diseño y evaluación de algoritmos de detección basados en el realce y la umbralización de los bordes. Analizaron dos tipos de detectores de bordes:

- Operadores “diferenciales”: operador de Sobel (Duda y Hart 1973)(página 271), operador de Prewitt (Prewitt 1970) y operador de Roberts (Roberts 1965).
- Operadores de “emparejamiento de plantillas”: incluye al operador de brújula (*compass operators*) (Prewitt 1970), al operador de Kirsch (1.971) y a los operadores de 3 y 5 niveles.

Se ha de indicar que Abdou y Pratt no indican el método de umbralización que aplicaron a los algoritmos de detección.

Características de las imágenes utilizadas

Utilizaron dos tipos de imágenes:

- Imágenes artificiales: imágenes con un borde central vertical o diagonal del tipo “rampa” situado sobre un fondo oscuro. A estas imágenes se les añadió ruido gaussiano aditivo de media cero de forma que la proporción entre la señal y el ruido, denotada por SNR (*signal to noise ratio*), fuera igual a 1, 10 ó 100. El valor de SNR fue definido como

$$SNR = \left(\frac{h}{\sigma} \right)^2 \quad (4.2)$$

donde h es el contraste del borde y σ es la desviación típica del ruido.

⁵La medida de Pratt es conocida como *F.O.M.: figure of merit*.

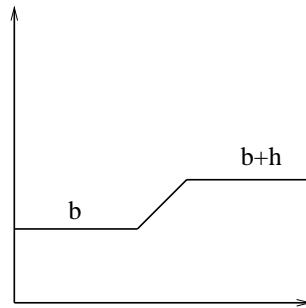


Figura 4.3: Estructura del borde de “rampa” utilizada en el estudio de Abdou y Pratt.

- Imágenes reales: se utilizaron tres imágenes diferentes, correspondientes a una mujer, a una pista de un aeropuerto y a un tanque.

Los autores no indican la definición espacial (número de puntos) ni la resolución digital (número de niveles de gris) de las imágenes.

Métodos de diseño y evaluación de los detectores de bordes

Los autores realizaron un estudio de los detectores de bordes que puede ser dividido en dos partes:

- Análisis teórico para el diseño de detectores de bordes.
- Evaluación comparativa de los detectores de bordes utilizando la medida de Pratt.

El análisis teórico estudiaba la sensibilidad de los detectores de bordes respecto de ciertas propiedades que deberían poseer:

- La amplitud estimada del borde no debe depender de su orientación.
- La medida de la orientación del borde no debe ser sesgada.
- Debe haber una rápida disminución de la estimación del gradiente del borde cuando la máscara del detector se aleja del borde central.

Los procedimientos teóricos para el diseño de detectores fueron dos:

- Procedimiento estadístico para la selección del umbral del detector de bordes: se determina el umbral analizando las probabilidades de detección correcta y de detección falsa de los bordes. Este método requiere un conocimiento previo del tipo de bordes y además presenta dos dificultades:

- Fiabilidad del modelo estocástico de borde.
- Problemas analíticos asociados con los complejos modelos de bordes tales como el ruido no gaussiano dependiente de la señal.
- Procedimiento de diseño de clasificación de patrones: la detección de bordes puede ser vista como un problema de reconocimiento de patrones o de clasificación.

Las probabilidades de detección correcta y detección falsa de los bordes, obtenidas analíticamente o experimentalmente, son indicadores del rendimiento útiles para los detectores de bordes. Sin embargo, estas funciones de probabilidad de detección no distinguen entre varios tipos de errores que pueden ser introducidos por un detector de bordes como los indicados en la figura 4.4.

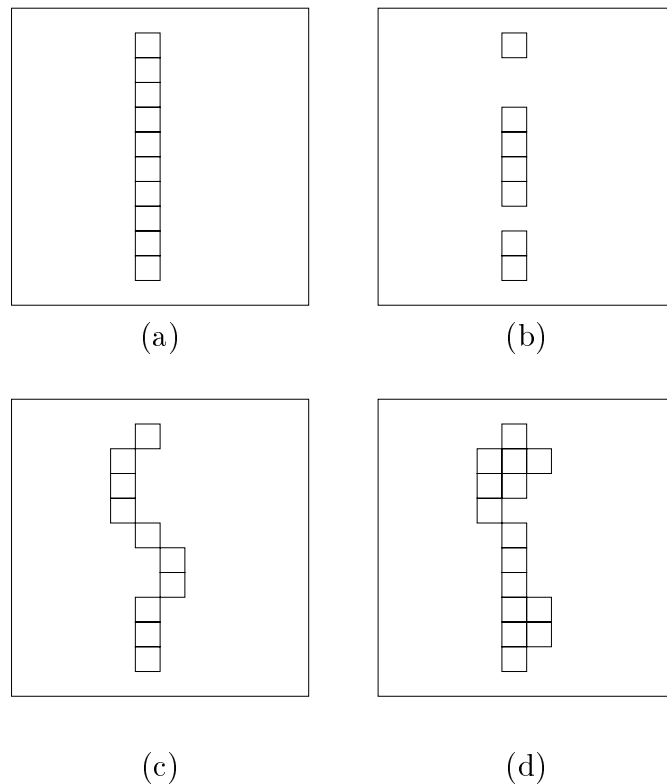


Figura 4.4: Tipos de errores en la detección de bordes: (a) borde ideal, (b) borde fragmentado, (c) borde desplazado y (d) borde engrosado.

Abdou y Pratt utilizaron la medida propuesta por Pratt (1.978) para evaluar la calidad de los detectores de bordes: desviación ponderada y normalizada de los puntos del borde detectados respecto del borde ideal:

$$F = \frac{1}{\text{máximo}\{I_I, I_A\}} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d(i)^2} \quad (4.3)$$

donde

- I_A es el número de puntos de borde detectados.
- I_I es el número de puntos del borde ideal.
- α es una constante de escala.
- $d(i)$ es la distancia entre un punto de borde y el borde ideal.

La constante α pondera los errores producidos por los bordes gruesos o desplazados. El valor de α elegido por Abdou y Pratt fue $\frac{1}{9}$.

Esta medida fue diseñada para evaluar el rendimiento de los detectores ante bordes verticales. Si se utilizan bordes diagonales, se debe utilizar el factor de normalización $\sqrt{2}$.

Finalmente, aplicaron los detectores seleccionados a las imágenes reales para contrastar los resultados obtenidos.

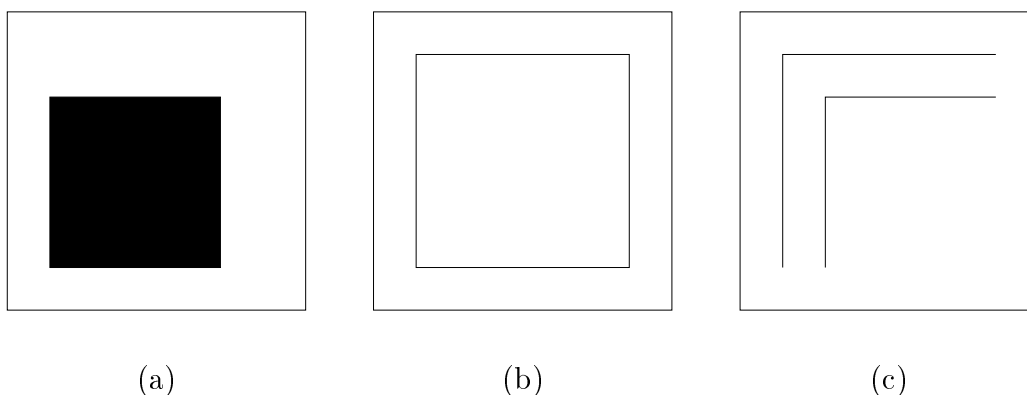


Figura 4.5: Dificultades de la medida de Pratt: (a) imagen con un objeto; (b) y (c) bordes detectados que tienen el mismo valor según dicha medida.

Peli y Malah (1.982) indicaron que la medida de Pratt no suministra suficiente información sobre el rendimiento de los detectores de bordes probados. Hay situaciones en las que valores altos de la medida de Pratt están asociados a bordes visualmente rotos. La razón es que dicha medida no tiene en cuenta la distribución de los puntos de borde detectados a lo largo del borde. Por ejemplo, si se aplican dos algoritmos de detección de bordes a la imagen de la figura 4.5(a) y se obtienen los bordes de las figuras 4.5(b) y (c), entonces los valores de la medida de Pratt son los mismos.

4.3.3 Uso de las componentes necesarias de un espacio vectorial de errores para la evaluación de los detectores de bordes

Descripción

Venkatesh y Kitchen (1.992) clasificaron los errores que pueden ocurrir durante la detección de bordes en cuatro tipos y a partir de ellos desarrollaron varias medidas de evaluación cuantitativa. Mostraron que las medidas definidas eran independientes y formaban un conjunto completo. Además, combinaron las medidas cuantitativas para desarrollar otra medida que tuviera en cuenta ciertas propiedades cualitativas.

Sus medidas de evaluación fueron aplicadas a algoritmos de detección clásicos de realce y umbralización, para lo cual utilizaron imágenes artificiales.

Detectores de bordes evaluados

Se seleccionaron cuatro detectores de bordes basados en el realce y la umbralización:

- Operador de Roberts (1.965).
- Operador de Sobel (1.978).
- Operador de Prewitt (1.970).
- Operador de la máscara de tres puntos (Venkatesh y Owens 1990).

El modo de aplicación de cada uno de los detectores de bordes fue el siguiente:

1. Aplicación de un operador de suavización⁶.
2. Aplicación del detector de bordes.
3. Escalamiento logarítmico y umbralización.

Utilizaron el siguiente escalamiento logarítmico debido a restricciones de implementación de los valores de los puntos de la imagen:

$$S = \begin{cases} 255 \frac{\log_{10}(E)}{\log_{10}(M)} & \text{si } E > 0 \\ 0 & \text{si } E = 0 \end{cases} \quad (4.4)$$

⁶Los autores no indican qué operador fue utilizado.

donde E denota el valor en un punto y M denota el valor máximo en la imagen.

Esta operación no tiene ningún efecto real sobre la umbralización, ya que “la toma de logaritmos” es una operación monótona.

El umbral fue elegido de forma que se optimizara la medida de cada detector.

Características de las imágenes

Se utilizaron tres conjuntos de imágenes artificiales:

- Imágenes ruidosas con bordes verticales (figura 4.6): cada imagen estaba dividida en tres partes: un rectángulo situado a la izquierda con un nivel de gris de 115, un rectángulo situado a la derecha con un nivel de gris de 140 y una línea vertical y central, cuyos puntos tenían un nivel de gris de 115. La línea vertical tenía el grosor de un punto. El modelo de borde obtenido es una “rampa” ligeramente asimétrica.

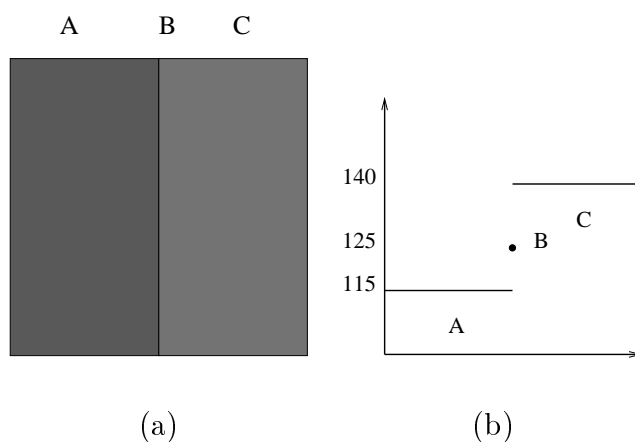


Figura 4.6: Borde vertical de Venkatesh y Kitchen: (a) imagen sin ruido y (b) nivel de gris de los puntos situados en una línea perpendicular al borde.

A estas imágenes se les añadió ruido gaussiano de media cero con diferentes valores de desviación típica. La proporción entre la señal y el ruido (SNR) fue definida usando el criterio de Pratt (1.978):

$$SNR = \left(\frac{h}{\sigma} \right)^2 \quad (4.5)$$

donde h es el contraste del borde (25 en este caso) y σ es la desviación típica del ruido. Para simular condiciones de bajo, medio y gran ruido, se consideraron los siguientes valores de SNR : 100, 14'70 y 2.

- Imágenes, sin ruido, con bordes orientados (figura 4.7): se generaron imágenes similares a las anteriores pero en las cuales el borde tenía las siguientes

pendientes: 0'5 (26'57°), 0'7 (34'99°) y 0'85 (40'36°). A estas imágenes no se les añadió ruido. Las imágenes tenían un triángulo inferior izquierdo con un nivel de gris de 110 y un rectángulo superior derecho con un nivel de gris de 140. El borde orientado fue generado utilizando el método propuesto por Abdou y Pratt (1.979) para simular el proceso de digitalización de una cámara: se considera una máscara de 3×3 puntos situada sobre un borde con una orientación arbitraria

$$y = mx + c \quad (4.6)$$

donde m es la pendiente del borde; el nivel de gris de un punto situado a la derecha del borde orientado como el de la figura 4.7(b) es proporcional al área del polígono formado por la intersección del borde orientado y la máscara. Por ejemplo, el nivel de gris del punto de la figura 4.7(b) sería

$$\frac{1}{2}(1 - r)30s + 110 \quad (4.7)$$

donde $1 - r = s$.

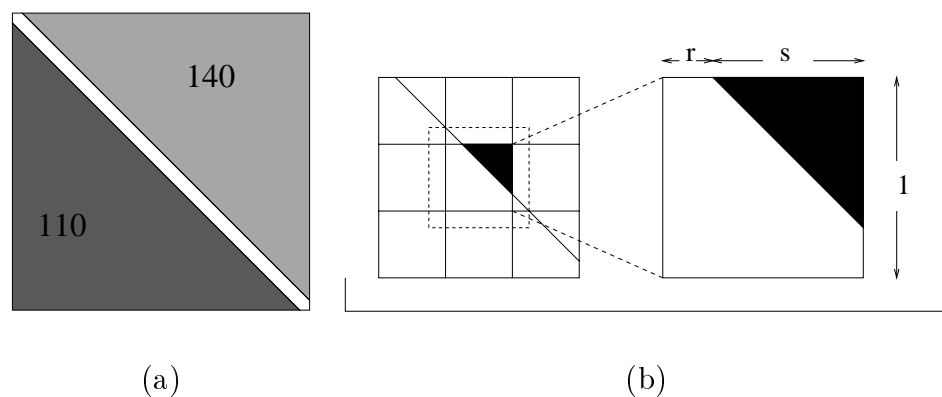


Figura 4.7: Borde orientado de Venkatesh y Kitchen: (a) imagen sin ruido y (b) proporciones utilizadas para calcular el nivel de intensidad de un punto por el que pasa el borde.

- Imágenes ruidosas con bordes orientados: se añadieron diferentes niveles de ruido gaussiano a la imagen que tenía un borde con pendiente $m = 0'7$ de forma que los valores de SNR fueran 144, 21 y 3.

Todas las imágenes artificiales estaban compuestas por 128×128 puntos.

Método de evaluación basado en las componentes necesarias

El método de evaluación de Venkatesh y Kitchen está basado en la integración de diferentes errores:

- Falsos negativos: no detección de puntos que pertenecen al borde. El efecto de este error es producir bordes rotos o discontinuos.
- Falsos positivos: detección de puntos que no pertenecen al borde.
- Detección múltiple: detección de más de un punto como respuesta a un único punto perteneciente al borde. Este error mide el grosor del borde detectado.
- Localización: error entre la posición del borde detectado y la posición del borde verdadero.

Para poder definir cuantitativamente la influencia de cada error es necesario introducir los conceptos de “tira de detección de un borde”. La figura 4.8 muestra las tiras de detección para los bordes verticales y orientados. La tira de detección indica la zona de la imagen en la que debe ser detectado un borde para que el funcionamiento del operador sea correcto. En el caso de una imagen que contenga un borde vertical en la columna p , la tira de detección se corresponde con las columnas pertenecientes al intervalo $[p - e, p + e]$, donde e generalmente vale 1. Análogamente se define la tira de detección de un borde orientado, que suele ser más ancha debido a la digitalización.

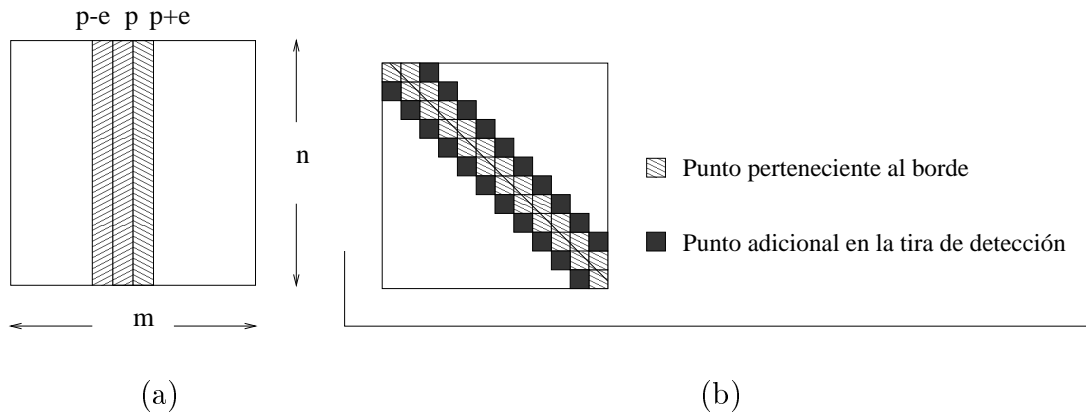


Figura 4.8: Detección de bordes (a) verticales y (b) orientados.

Las siguientes definiciones corresponden al borde vertical de la figura 4.8, siendo similares las del borde orientado. Sea n el número de filas y m el número de columnas de la imagen. Sea A_1 el área que no contiene a la tira de detección. Si $e = 1$,

$$A_1 = n(m - 3) \quad (4.8)$$

Sea A_2 el área de la tira de detección:

$$A_2 = 3n \quad (4.9)$$

Las medidas asociadas a cada uno de los errores descritos anteriormente son las siguientes:

- Falso negativo

$$FN = \frac{f_{nd}}{n} \quad (4.10)$$

donde f_{nd} es el número de filas de la tira de detección en las que no se ha detectado ningún punto del borde. El denominador n asegura que esta medida sea independiente del tamaño de la imagen.

- Falso positivo:

$$FP = \frac{p_f}{A_1} \quad (4.11)$$

donde p_f es el número de puntos detectados fuera de la tira de detección.

- Detección múltiple:

$$MD = \frac{p_{md}}{(A_2 - n)} \quad (4.12)$$

donde p_{md} es el número de múltiples bordes dentro de la tira de detección.

- Localización:

$$LOC = \frac{\sum |p_d - p_v|}{t} \quad (4.13)$$

donde p_d es la posición del borde detectado, p_v es la posición del borde verdadero y t denota el número de filas donde hay puntos de borde. La suma abarca a todos los puntos detectados. Si hay más de un punto detectado en una fila, el error de localización es calculado considerando el punto más cercano al borde verdadero. Se realiza, pues, el análisis en el mejor caso posible, lo cual puede ser considerado como un límite superior al rendimiento del detector o un límite inferior sobre el error.

Una vez definidos los errores de detección, Venkatesh y Kitchen comprobaron que los tipos de errores considerados eran independientes entre sí, es decir, la medida de un tipo de error puede cambiar sin que cambien las medidas de los otros tipos de error. La figura 4.9 muestra ejemplos ilustrativos de la independencia de los tipos de error.

Los cuatro tipos de error pueden ser considerados como componentes de una base de “un espacio vectorial de errores”, de forma que a partir de ellos se puedan definir otras medidas de error. Venkatesh y Kitchen definen la siguiente “medida de fracaso” (*failure measure*)

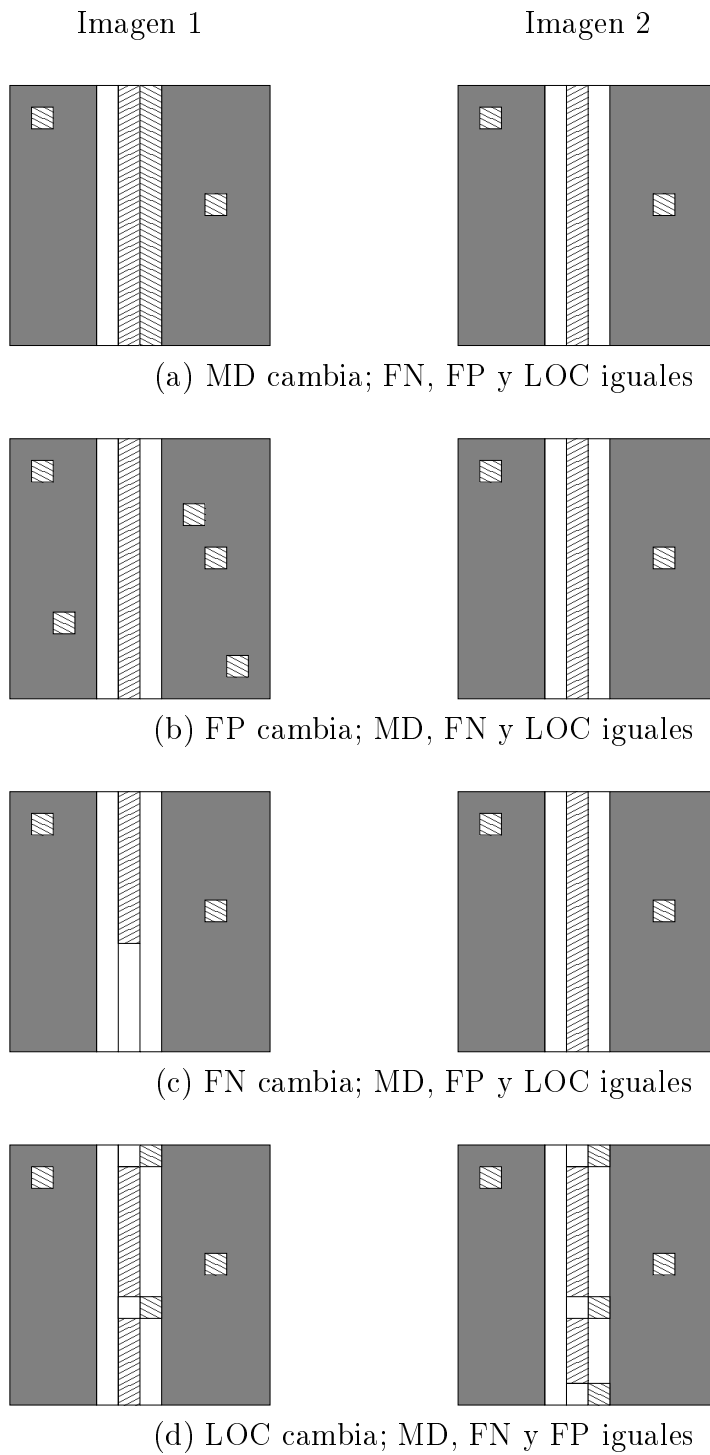
$$FM = \alpha FP + \beta FN + \gamma MD + \delta LOC \quad (4.14)$$

donde

$$\alpha + \beta + \gamma + \delta = 1.0 \quad (4.15)$$

y

$$\alpha, \beta, \gamma, \delta \geq 0.0 \quad (4.16)$$



▨ Puntos detectados dentro y fuera de la tira de detección

■ Área fuera de la tira de detección

Figura 4.9: Independencia de los tipos de errores: en cada caso sólo cambia la medida de un tipo de error, permaneciendo los demás iguales.

Tabla 4.1: Coeficientes de las medidas de los tipos de error.

FM	α	β	γ	δ
1	0'4	0'4	0'15	0'05
2	0'4	0'3	0'25	0'05
3	0'3	0'4	0'25	0'05
4	0'3	0'4	0'2	0'1
5	0'4	0'3	0'15	0'15
6	0'4	0'3	0'2	0'1

La tabla 4.1 muestra los distintos coeficientes utilizados por Venkatesh y Kitchen para definir seis medidas de fracaso diferentes.

Venkatesh y Kitchen destacan que la medida de fracaso propuesta tiene una gran flexibilidad al permitir la elección de los coeficientes que se consideren más adecuados, ya que, por ejemplo, en algunas aplicaciones es más importante la detección de los bordes que su grosor, mientras que en otras se exige que los bordes detectados tengan la anchura de un punto.

No obstante, los autores indican que las seis medidas de fracaso utilizadas clasificaron a los detectores de bordes seleccionados de forma muy similar. En conjunto, el operador de la máscara de tres puntos es el que obtuvo mejor evaluación.

4.4 Evaluación de algoritmos de detección de bordes en imágenes en color

Los estudios comparativos de algoritmos de detección de bordes en imágenes en color son muy escasos, sobre todo si se comparan con los realizados con los algoritmos diseñados para imágenes monocromáticas. Sólo se ha tenido conocimiento de cuatro estudios que evalúan exclusivamente algoritmos de detección de bordes en color (Koschan 1995; Novak y Shafer 1987; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999). Otros estudios han evaluado algoritmos de detección de bordes junto con algoritmos de segmentación orientados a regiones (Allen y Huntsberger 1989; Gauch y Hsia 1992).

Las tablas 4.6 - 4.8 resumen las propiedades básicas de los estudios comparativos de detectores de bordes en color, los cuales son descritos a continuación.

4.4.1 Estudio de las versiones de color del operador de Canny

Descripción

Novak y Shafer (1.987) realizaron un estudio subjetivo de diferentes versiones del algoritmo de Canny (1.986) que habían sido adaptadas para su aplicación a imágenes en color.

Detectores de bordes evaluados

Novak y Shafer indicaron que cualquier método de detección monocromática de bordes puede ser extendido a imágenes en color. Si un detector monocromático consta de n pasos o fases entonces un detector en color puede ser obtenido en la fase k de la siguiente forma:

1. En las fases 1 a k , se aplica el detector monocromático a cada una de las componentes de la imagen en color.
2. En la fase k , se utiliza un operador para combinar los resultados de cada una de las componentes.
3. En las fases $k + 1$ a n , se continúa aplicando el detector monocromático al resultado de la operación de combinación.

Generalmente, los operadores de combinación son distancias métricas, como la norma L_1 (suma de los valores absolutos), la norma L_2 (distancia euclídea) o la norma L_∞ (máximo de los valores absolutos).

El detector de bordes en color puede ser caracterizado por k/i , donde k es la fase en la que se produce la combinación e i es el índice de la norma utilizada como operador de combinación.

Esta técnica de extensión de detectores monocromáticos fue aplicada al operador de Canny (1.986), para lo cual se estableció la siguiente numeración de sus fases:

- (I) Obtención de las derivadas direccionales en cada una de las componentes de color de la imagen.
- (II) Cálculo de la magnitud y de la dirección del gradiente.
- (III) Identificación de los bordes mediante la supresión de los puntos que no son máximos locales.

La variante en color $I/2$ del operador de Canny consiste en

- la obtención de forma independiente de las derivadas direccionales de cada componente de color,
- la combinación de estas derivadas mediante la distancia euclídea para obtener la magnitud total del cambio de color
- y la utilización de este valor para los cálculos restantes.

Las versiones $0/1$ y $0/2$ de este operador de color de Canny sólo operan sobre una imagen monocromática y aplican la versión original del operador de Canny. El operador $I/1$ es similar que el operador $0/1$ puesto que las derivadas direccionales son calculadas sobre una transformación lineal de la imagen. Finalmente indicar que el operador de color que realizara la combinación en la fase IV obtendría los bordes de cada componente de forma independiente y después los combinaría para generar el mapa de bordes final.

Novak y Shafer compararon la versión original del operador de Canny con todas las versiones no triviales de los operadores de color obtenidas y con el detector de bordes en color basado en el concepto de gradiente multidimensional que utiliza la matriz jacobiana (Lee y Cok 1991).

Imágenes utilizadas

Se utilizaron imágenes reales correspondientes a edificios de una ciudad. Los autores no indican la definición espacial ni la resolución digital de dichas imágenes. Las imágenes estaban definidas en el espacio de color RGB .

Metodología de evaluación

Se realizó una evaluación subjetiva que consistió en una comparación visual de los resultados obtenidos por los diferentes operadores.

Conclusiones

Las principales conclusiones citadas por los autores son:

- Los bordes obtenidos de la imágenes en color son consistentemente mejores que los de las imágenes monocromáticas, aunque la mayoría (alrededor del 90%) son los mismos.

- El operador que obtiene mejores resultados es la versión de color II/∞ del operador de Canny, es decir, el que calcula la magnitud y la dirección del gradiente independientemente para cada componente de color y después selecciona el borde con máxima magnitud.
- El operador I/∞ también tiene un buen rendimiento, pero sus resultados no superan a los del operador II/∞ . Sin embargo, el operador I/∞ es más rápido que el operador II/∞ puesto que la combinación se efectúa antes.
- El operador II/∞ produce casi exactamente los mismos bordes que el operador basado en el gradiente multidimensional.

4.4.2 Estudio realizado por Allen y Huntsberger

Descripción

Allen y Huntsberger (1.989) realizaron un estudio comparativo de dos algoritmos de detección de bordes y de dos algoritmos de segmentación orientados a regiones.

Para evaluar los algoritmos, se utilizaron imágenes reales y artificiales con diferentes niveles de ruido.

El estudio pretendía analizar la influencia del ruido sobre los algoritmos y determinar qué componentes de los diferentes espacios de color era más útil.

Algoritmos evaluados

Se evaluaron cuatro algoritmos de segmentación de imágenes en color:

- Algoritmos de detección de bordes:
 - Algoritmo de Nevatia (1.977): este método es una generalización a imágenes multiespectrales del operador de Hueckel (1.971, 1.973).
 - Operador *DOOG* (*difference of offset Gaussians*): es un operador del tipo “cruce por cero” rotacionalmente invariante (Young 1986b).
- Algoritmos orientados a regiones:
 - Algoritmo de Ohta (1.980,1.985).
 - Método de segmentación iterativo *fuzzy c-means* (Huntsberger et al. 1985; Jayaramamurthy y Huntsberger 1985).

Para poder comparar los cuatro algoritmos entre sí, se consideró que los métodos orientados a las regiones definían los bordes o contornos de los objetos a partir de las regiones en que dividían a las imágenes.

Hay que destacar que cada uno de estos algoritmos utiliza un espacio de color diferente⁷:

- El algoritmo de Nevatia hace uso de las siguientes componentes: Y (intensidad), T_1 (rojo normalizado) y T_2 (verde normalizado).
- El operador *DOOG* utiliza dos componentes obtenidas del modelo *RGBY*⁸: $+R - G$ y $+B - Y$.
- Ohta (1.985) utiliza su propio espacio de color: I (intensidad), $I'_2 = R - B$ y $I'_3 = (2G - R - B)/2$.
- El algoritmo *fuzzy c-means* puede ser aplicado a diferentes espacios de color. Sus autores utilizaron el espacio de color *RGB*, el espacio de color de Ohta y el modelo *CIE XYZ*.

Características de las imágenes utilizadas

Se utilizaron imágenes reales y artificiales con un tamaño de 240×256 puntos. Cada punto de la imagen tomaba valores entre 0 y 255 en cada plano del espacio de color *RGB*. Se obtuvieron variantes de estas imágenes definidas en los espacios de color *Yrg*, el espacio de color de colores contrarios $R - G$ y $Y - B$ y en el espacio de color propuesto por Ohta (1.980,1.985).

Todas las imágenes artificiales estaban compuestas por 16 rectángulos de 60×64 puntos cada uno. Se generaron tres tipos de imágenes artificiales:

- Imágenes de tablero de ajedrez:
 - Se generó una única imagen sin ruido y con alto contraste. Se seleccionaron aleatoriamente valores de *RGB* para crear un contraste de 2:1 entre los rectángulos contiguos.
 - Imágenes de igual iluminación: se generó una imagen con dos colores “isoluminantes” de forma que los valores de intensidad de todos los rectángulos fueran iguales. Los valores de las componentes de *RGB* fueron creadas aleatoriamente con la restricción de preservar la “isoluminancia”. A partir de esta imagen original se generaron 7 imágenes con diferentes niveles de ruido.
- Imagen *quilt*⁹: todos los rectángulos tenían igual iluminación (intensidad) pero diferente contraste de colores. Los valores de las componentes de color

⁷Véase el apéndice A Color.

⁸*R*: red (rojo), *G*: green (verde), *B*: blue (azul) e *Y*: yellow (amarillo)

⁹Literalmente, “colcha”.

fueron generados aleatoriamente. La figura 4.10 muestra un ejemplo de la estructura de este tipo de imágenes.

R = 109	R = 165	R = 183	R = 152
G = 133	G = 120	G = 121	G = 150
B = 255	B = 176	B = 121	B = 55
R = 80	R = 169	R = 15	R = 163
G = 197	G = 144	G = 183	G = 123
B = 2	B = 42	B = 245	B = 165
R = 78	R = 13	R = 68	R = 255
G = 167	G = 218	G = 154	G = 75
B = 163	B = 71	B = 255	B = 175
R = 135	R = 56	R = 223	R = 255
G = 145	G = 179	G = 75	G = 59
B = 131	B = 159	B = 255	B = 255

Figura 4.10: Estructura de una imagen *quilt*.

Además de la imágenes artificiales, se utilizaron 5 imágenes reales, aunque, debido a limitaciones de espacio, los autores sólo muestran dos: una imagen de una casa y una imagen de una carretera con dos coches.

Metodología de evaluación

Allen y Huntsberger utilizaron dos métodos de evaluación:

- Evaluación objetiva: el rendimiento de los algoritmos sobre imágenes artificiales fue evaluado utilizando una variante de la medida propuesta por Pratt (ecuación 4.3). Para cada algoritmo y cada tipo de imagen, se promediaron los valores de la medida de Pratt obtenidos.
- Evaluación subjetiva: los resultados obtenidos al aplicar los algoritmos a las imágenes reales fueron analizados visualmente, comprobando qué bordes habían sido detectados y qué componente de color era la que más información suministraba al proceso de detección.

Conclusiones

A partir de los resultados mostrados por Allen y Huntsberger, se pueden obtener las siguientes conclusiones.

- El rendimiento de los algoritmos depende del tipo de imágenes:

- Las pruebas con la imagen *quil* reflejan que los métodos orientados a regiones superan a los algoritmos de detección de bordes cuando el porcentaje de ruido es bajo. Por el contrario, si los niveles de ruido superan el 20%, el rendimiento de los algoritmos de detección de bordes es superior al de los métodos orientados a regiones.
- Las pruebas con las otras imágenes indican que el rendimiento del algoritmo *fuzzy c-means* es el que obtiene los mejores resultados. El algoritmo de Ohta tiene un rendimiento superior al de los algoritmos de detección de bordes cuando el ruido es inferior al 15%.
- Algunos de los espacios de color poseen componentes que son más útiles que las demás en el proceso de detección de bordes:
 - Para el algoritmo de Nevatia, la componente Y (intensidad).
 - Para el operador *DOOG*, la componente $+B - Y$.

4.4.3 Comparación de tres algoritmos de segmentación en cuatro espacios de color

Descripción

Gauch y Hsia (1.992) compararon tres algoritmos de segmentación que fueron aplicados a imágenes definidas sobre cuatro espacios de color diferentes. Su objetivo era analizar el rendimiento de los algoritmos y comprobar qué espacio de color era más útil.

Detectores de bordes evaluados

Se compararon tres algoritmos (Gauch y Hsia 1992):

- Algoritmo de detección de bordes basado en los cruces por cero de la derivada direccional de la magnitud del gradiente en la dirección del gradiente.
- Algoritmo de segmentación basado en el crecimiento de regiones.
- Algoritmo de segmentación basado en la división y fusión de regiones.

Se consideró que el algoritmo de detección de bordes definía las regiones de las imágenes en función de los bordes que detectaba.

Imágenes utilizadas

Se utilizaron los espacios de color *RGB*, *HLS*, *YIQ* y *Lab* para obtener diferentes imágenes de una escena real (una carretera) y de una escena compuesta por 9 cuadrados, cada uno de un color diferente. A las imágenes se les añadió ruido gaussiano.

Metodología de evaluación

Los algoritmos fueron evaluados objetivamente, para lo cual se hizo uso de imágenes de referencia en las que el mapa de bordes fue definido “manualmente”. En dichas imágenes se indicaban las regiones en las que debería dividirse la imagen.

Para valorar la calidad de los algoritmos, se calcularon las siguientes cantidades para cada región:

- Proporción entre el número de puntos “correctamente” detectados en la región respecto del total de puntos detectados en dicha región:

$$\frac{TP}{TP + FP} \quad (4.17)$$

- Proporción entre el número de puntos “correctamente” detectados en la región respecto del total de puntos que deberían haber sido detectados correctamente en dicha región:

$$\frac{TP}{TP + FN} \quad (4.18)$$

donde

- *TP* es el número de puntos correctamente detectados o verdaderos positivos (*true positive*),
- *FP* es el número de puntos incorrectamente detectados o falsos positivos (*false positive*),
- y *FN* es el número de puntos incorrectamente no detectados o falsos negativos (*false negative*).

Conclusiones

Entre las conclusiones citadas por los autores, se han de destacar las siguientes:

- Ningún espacio de color fue superior a los demás. Existe cierta dependencia entre el contenido de la imagen y el espacio de color utilizado:
 - Para la escena de la carretera, los espacios de color *HLS* y *Lab* superan a los espacios de color *RGB* e *YIQ*.
 - Para la escena de los cuadrados, los espacios de color *RGB* e *YIQ* superan a los espacios de color *HLS* y *Lab*.
- El algoritmo de división y fusión de regiones fue el que obtuvo los resultados más satisfactorios, independientemente del contenido de las imágenes y del espacio de color utilizado.

4.4.4 Comparación realizada por Trahanias y Venetsanopoulos

Descripción

Trahanias y Venetsanopoulos (1.993,1.996) utilizaron unos operadores estadísticos de ordenamiento de vectores para diseñar nuevos detectores de bordes en color. Se utilizó la medida de Pratt (1.978) para comparar cuantitativamente los nuevos detectores entre sí y determinar cual de ellos tenía un mejor rendimiento. Posteriormente, el detector de bordes que obtuvo mejores resultados fue comparado subjetivamente con otros detectores de bordes de diferentes tipos.

Detectores de bordes evaluados

Los detectores de bordes evaluados fueron:

- Detectores basados en operadores estadísticos de ordenamiento de vectores (Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996):
 - VR (*Vector Range*).
 - MVR (*Minimun Vector Range*).
 - VD (*Vector Dispersion*).
 - MVD (*Minimun Vector Dispersion*).
- Detector basado en la suma de los vectores gradientes de las componentes de la imagen en color (Pratt 1991).
- Detector basado en el vector gradiente multidimensional (Zenko 1986).
- Detector basado en las derivadas de segundo orden (Cumani 1991).
- Detector basado en la entropía (Shiozaki 1986).

Imágenes utilizadas

Se utilizaron dos tipos de imágenes:

- Imágenes artificiales con las siguientes características:
 - Contienen bordes verticales, horizontales y diagonales.
 - Hay bordes donde todas las componentes de color, o dos de las componentes de color o una de las componentes de color cambian.
 - Contienen áreas negras (todas las componentes de color son cero), blancas (todas las componentes de color tienen el valor máximo, 255) y grises (todas las componentes de color tienen el mismo valor).
 - Contienen áreas isoluminantes (áreas con idéntica iluminación) y áreas no isoluminantes.
- Imágenes reales de muy diferentes tipos: desde imágenes de rostros humanos hasta imágenes de escenas exteriores. En el artículo sólo se muestran los resultados de una imagen correspondiente a un avión que sobrevuela una zona montañosa.

Las imágenes fueron corrompidas mediante la adición de diferentes niveles de ruido gaussiano, doble exponencial e impulsivo. Para cada tipo de ruido, se realizaron dos tipos de experimentos:

- Se consideró que el ruido de cada componente de color era generado mediante un proceso independiente.
- Se consideró que el ruido de las componentes estaba correlado.

Metodología de evaluación

Se realizaron dos estudios comparativos:

- Estudio cuantitativo de los diferentes detectores de bordes basados en el ordenamiento de vectores. Los detectores de bordes fueron aplicados a las imágenes artificiales con ruido. Para comparar su rendimiento, se utilizó la medida propuesta por Pratt (1978) (véase la ecuación 4.3).
- Estudio cualitativo del detector MVD (detector de bordes basado en el ordenamiento de vectores que obtuvo mejores resultados en el estudio anterior), el detector basado en la suma de los gradientes (Pratt 1991), el detector basado en el vector gradiente (Zenko 1986), detector basado en los operadores de segundo orden (Cumani 1991) y el detector basado en la entropía

(Shiozaki 1986). Se utilizó la medida de Pratt para evaluar el rendimiento de estos algoritmos ante imágenes artificiales. También se realizó una evaluación subjetiva utilizando imágenes reales.

Conclusiones

Entre las conclusiones indicadas por los autores, se pueden destacar:

- El detector de bordes MVD es el que tiene el mejor rendimiento entre todos los detectores basados en el ordenamiento de vectores.
- El detector de bordes MVD supera a los demás detectores de bordes evaluados, ya que es más robusto en presencia de ruido, los bordes que detecta están más adelgazados y es menos sensible a pequeños cambios de textura.

4.4.5 Comparación realizada por Koschan

Descripción

El estudio de Koschan (1.995) trataba de alcanzar los siguientes objetivos:

- Comparar detectores de bordes en imágenes en color con variantes multiespectrales de detectores de bordes en imágenes monocromáticas.
- Mostrar que las imágenes en color aportan más información que las imágenes monocromáticas.

Koschan utilizó imágenes artificiales, a las que les añadió ruido gaussiano, e imágenes reales para realizar una evaluación subjetiva de los detectores de bordes.

Detectores de bordes evaluados

Los detectores de bordes evaluados se pueden clasificar en dos grupos:

- Variantes de detectores de bordes en imágenes monocromáticas: se utilizaron los siguientes operadores:
 - Operador de Sobel (1.970, 1978). Para poder ser generalizado a imágenes en color, este operador se aplicó a cada uno de los planos de la imagen. La suma de los cuadrados de las convoluciones obtenidas fue utilizada como aproximación de la magnitud del gradiente en cada plano.

Un punto fue considerado como punto de borde si la media de las magnitudes de los gradientes superaba un umbral preestablecido.

- Operador de Laplace: se utilizó la máscara de convolución de la figura 4.11 que fue aplicada a cada uno de los planos de color de cada imagen. Un punto fue considerado de borde si se producía en él un “cruce por cero” en alguno de los planos.

$$M_{Laplace} = \begin{bmatrix} 10 & 22 & 10 \\ 22 & 128 & 22 \\ 10 & 22 & 10 \end{bmatrix}$$

Figura 4.11: Máscara de convolución del operador de Laplace.

- Operador del “sombrero mejicano” (Marr y Hildreth 1980):

$$-\nabla^2 G(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6} e^{(-\frac{x^2+y^2}{\sigma^2})} \quad (4.19)$$

Se generaron máscaras de convolución utilizando la ecuación 4.4.5. El proceso para determinar si un punto era de borde fue similar al utilizado con el operador de Laplace.

- Detectores de bordes en imágenes en color basados en los “cruces por cero” de la segunda derivada direccional:
 - Operador de Cumani (Cumani 1991): se utilizaron diferentes versiones de este operador, ya que se usaron máscaras de convolución de diversos tamaños. Además, Koschan introdujo una variante en el operador de Cumani que permitía disminuir el tiempo de ejecución.
 - Operador de Alshatti - Lambert: este operador es una variante del operador de Cumani (Alshatti y Lambert 1993). También se utilizó la mejora introducida por Koschan en el operador de Cumani.

Imágenes utilizadas

Se utilizaron imágenes en color artificiales y reales con las siguientes características:

- Las imágenes artificiales representaban pequeñas líneas, esquinas, uniones y esferas de diferentes colores. A estas imágenes se les añadió ruido gaussiano con desviación típica $\sigma = 10.0$ y $\sigma = 25.0$.
- Se utilizaron 7 tipos de imágenes reales representando escenas de interiores que tenían diferentes niveles de complejidad y condiciones de luz.

El espacio de color utilizado fue *RGB*. A partir de las imágenes en color se generaron imágenes monocromáticas promediando los valores de cada plano de color. Estas imágenes monocromáticas sólo fueron utilizadas por el operador de Cumani.

Metodología de evaluación

Todos los detectores de bordes seleccionados fueron aplicados a las imágenes en color artificiales y reales, realizando una evaluación visual subjetiva en la que se valoraba el grosor de los bordes, la sensibilidad al ruido, los bordes no detectados, los falsos puntos de borde, etc.

Además, el operador de Cumani fue aplicado a las imágenes monocromáticas para contrastar sus resultados con los obtenidos con imágenes en color.

Conclusiones

La principal conclusión del estudio de Koschan es la siguiente: el uso de imágenes en color permite obtener mejores resultados que con imágenes monocromáticas.

Además, Koschan destacó que los mejores resultados fueron obtenidos por los operadores de Cumani y Alshatti - Lambert, y que estos resultados mejoraban a medida que aumentaba el tamaño de las máscaras de convolución utilizadas.

4.4.6 Estudio comparativo realizado por Zhu, Platanioitis y Venetsanopoulos

Descripción

Este estudio comparativo examina diferentes tipos de detectores de bordes en imágenes en color, para lo cual utiliza tanto métodos de evaluación cuantitativa como subjetiva (Zhu et al. 1999).

Detectores de bordes evaluados

Se examinaron veinticuatro detectores de bordes que pueden ser clasificados en dos grupos (Zhu et al. 1999):

- Detectores de bordes basados en el ordenamiento de vectores: esta clase de detectores de bordes en color está caracterizada por el uso de combinaciones

lineales de vectores ordenados. Se pueden obtener diferentes detectores modificando los conjuntos de coeficientes. Doce operadores de esta clase fueron comparados en este estudio.

- Detectores basados en las diferencias de vectores: los operadores de esta clase puede ser considerados como operadores basados en la primera derivada. Se pueden obtener diferentes operadores modificando la forma de efectuar las diferencias entre los vectores asociados a cada punto de la imagen. En este estudio se examinaron doce detectores de bordes de esta clase.

Para poder contrastar la calidad de estos detectores de bordes en color, se utilizó, además, una extensión para imágenes en color del operador de Sobel.

Características de las imágenes

Se utilizaron imágenes en color definidas sobre el espacio *RGB*. No se indica la definición espacial ni la resolución digital.

Las imágenes eran de dos tipos:

- Imágenes artificiales: se generaron imágenes con diferentes tipos de bordes:
 - Bordes verticales, horizontales y diagonales.
 - Bordes suaves y abruptos.
 - Bordes provocados por la variación en una, dos o tres de las componentes del espacio del color.
 - Bordes provocados por áreas con igual o desigual iluminación.

A estas imágenes no se les añadió ruido.

- Imágenes reales sin ruido y con ruido mixto. Cada imagen fue corrompida con un 4% de ruido impulsivo y un 30% de ruido gaussiano. En ambos casos, la varianza fue de un 5%. Las imágenes reales contenían rostros humanos o escenas al aire libre.

Los autores no indican cuantas imágenes de cada tipo fueron utilizadas en las pruebas experimentales.

Metodología de evaluación utilizada

Se realizaron tres tipos de evaluaciones:

- Evaluación probabilística: los detectores de bordes fueron aplicados a las imágenes artificiales, que no contenían ruido. Los bordes detectados en las imágenes resultantes eran comparados con los bordes predefinidos, calculándose los siguientes valores:
 - Porcentaje de puntos de borde detectados (*hit ratio*).
 - Proporción entre el número de bordes falsos detectados y el número total de puntos de borde predefinidos (*fault ratio*).
- Rendimiento en presencia de ruido: los detectores de bordes fueron aplicados tanto a las imágenes reales sin ruido como a las corrompidas con ruido mixto. Los bordes detectados en las imágenes corrompidas fueron comparados con los detectados en las imágenes originales. El rendimiento del ruido fue medido en función de los valores de los “picos” de la proporción entre la señal y el ruido (*PSNR*).
- Evaluación subjetiva: los detectores de bordes fueron aplicados a una colección de imágenes reales y artificiales y los bordes detectados fueron examinados visualmente.

Conclusiones

Algunas de las conclusiones citadas por los autores son:

- Tanto los detectores de bordes basados en el ordenamiento de vectores como los basados en las diferencias de vectores son efectivos y eficientes.
- En ambos casos, las variantes de estos tipos de detectores con mayor complejidad computacional tienen mejores rendimientos en presencia de ruido.
- Las evaluaciones numérica y subjetiva ponen de manifiesto la superioridad de los detectores de bordes basados en la diferencia de vectores que incluyen una fase previa de filtraje adaptativo. Entre este tipo de detectores, destacan aquéllos que sólo realizan una exploración horizontal y vertical. Esto es debido a los siguientes hechos:
 - La visión humana es más sensible a los bordes verticales y horizontales que a los demás.
 - Las diferencias verticales y horizontales de vectores pueden detectar la mayoría de los bordes diagonales, reduciendo el grosor de estos bordes.
 - Su coste computacional es ligeramente inferior que el requerido por los detectores que exploran más direcciones.

Tabla 4.2: Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (primera parte).

Autor	Propósito	Tipo de evaluación	Tipo de imagen	Tipo de borde	Tipo de detector
(Fram y Deutsch 1975)	Evaluar la capacidad de detección y localización en presencia de ruido	Cuantitativa	Artificial y real con ruido gaussiano	Borde vertical de tipo “rampa”	Ajuste paramétrico Máscaras gaussianas
	Medir la sensibilidad a los cambios de orientación de los bordes	Cuantitativa y Cualitativa	Artificial	Borde orientado de tipo “rampa”	Diferencias locales
(Bryant y Boul-din 1979)	Evaluación absoluta y relativa de la capacidad de detección	Cuantitativa	Real con mapa de bordes definido manualmente	Bordes largos e irregulares de una escena rural	Realce y umbralización
(Abdou y Pratt 1979)	Evaluar la influencia del ruido y la orientación en la detección de los bordes	Teórica y cuantitativa	Artificial con ruido gaussiano	Borde vertical y diagonal de tipo “rampa”	Realce y umbralización
	Valorar los bordes detectados	Cualitativa	Real		
(Panda y Dubitzki 1979)	Análisis estadístico	Teórica y cuantitativa	Artificial y real	Aleatorio	Laplacianos Diferencias locales
(Kitchen y Rosenfeld 1981)	Evaluar la continuidad y la delgadez de los bordes detectados	Cuantitativa de bondad	Artificial con ruido gaussiano	Borde vertical de tipo “escalón” y circular de tipo “rampa”	Realce y umbralización

Tabla 4.3: Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (segunda parte).

Autor	Propósito	Tipo de evaluación	Tipo de imagen	Tipo de borde	Tipo de detector
(Peli y Malah 1982)	Comparar medidas de evaluación	7 cuantitativas y 3 cualitativas	Artificial con ruido gaussiano y con ruido de “sal y pimienta”	Bordes verticales, horizontales y circulares de tipos “escalón” y “rampa”	Realce y umbralización
	Comparar su nuevo método de detección de bordes	Cuantitativa y cualitativa	Artificial con ruido gaussiano y real sin ruido	Bordes verticales y horizontales de tipo “escalón”	Realce y umbralización Cruces por cero
(Haralick 1984)					
(Lyvers y Mitchell 1988)	Evaluar la sensibilidad a la traslación y rotación de los bordes	Teórica	Artificial con ruido gaussiano y artificial sin ruido	Bordes orientados de tipo “escalón”	Realce y umbralización Cruces por cero Ajuste paramétrico
(Owens y Venkatesh 1989)	Comprobar si los detectores poseen la propiedad de la proyección	Teórica y cualitativa	Artificial y real	Bordes verticales y horizontales de tipo “escalón”	Realce y umbralización Cruces por cero Energía
(Bernsen 1991)	Evaluar la capacidad de detección	Cuantitativa	Artificial	Bordes verticales y horizontales de tipo “escalón”	Realce y umbralización Ajuste paramétrico Cruces por cero Supresión de no máximos locales
	Comprobar la localización de patrones prefijados	Cualitativa	Real		

Tabla 4.4: Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (tercera parte).

Autor	Propósito	Tipo de evaluación	Tipo de imagen	Tipo de borde	Tipo de detector
(Ramesh y Haralick 1992a)	Evaluar la capacidad de detección	Teórica	Artificial con ruido gaussiano	Borde de tipo "rampa"	Ajuste paramétrico Morfológico
	Examinar los bordes detectados	Cualitativa	Real		
(Spreuwers y van der Heijden 1992)	Presentar un método basado en la media ponderada de varios errores	Cuantitativa	Artificial con ruido gaussiano	Bordes orientados de tipo "escalón"	Realce y umbralización Cruces por cero Supresión de no máximos locales
(Venkatesh y Kitchen 1992)	Presentar un método basado en el análisis combinado de varios errores	Cuantitativa	Artificial con ruido gaussiano y artificial sin ruido	Borde vertical de tipo "escalón" y orientados de tipo "rampa"	Realce y umbralización
(Strickland y Cheng 1993)	Presentar un método basado en la suma ponderada de seis medidas	Cuantitativa	Artificial con ruido gaussiano	Borde vertical de tipo "escalón"	Realce y umbralización Cruces por cero
(Ziou y Fabre 1994; Ziou et al. 1994)	Evaluar la influencia de la orientación de los bordes	Teórica	Artificial con ruido gaussiano	Borde orientado de tipo "escalón"	Supresión de no máximos locales
(Kamungo et al. 1995)	Evaluar la sensibilidad de los detectores a los cambios de orientación y contraste de los bordes	Cuantitativa	Artificial con ruido gaussiano	Bordes verticales y diagonales de tipo "escalón"	Detectores de líneas
(Salotti et al. 1996)	Evaluar la capacidad de detección de seis tipos de bordes	Cuantitativa	Real con mapa de bordes definido de forma manual o automática	Bordes de tipo "escalón" y "línea".	Supresión de no máximos locales

Tabla 4.5: Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (cuarta parte).

Autor	Propósito	Tipo de evaluación	Tipo de imagen	Tipo de borde	Tipo de detector
(Palmer et al. 1996)	Evaluar la detección de líneas	Cuantitativa	Artificial con ruido gaussiano y real sin ruido	Bordes orientados de tipo “escalón” y bordes de escenas reales	Detector de líneas
(Zhu 1996)	Presentar tres métodos para evaluar la conectividad y la uniformidad de la anchura de los bordes	Cuantitativas de bondad	Imagen binaria artificial con ruido e imagen binaria real sin ruido	Bordes circulares de imágenes binarias	No evalúa ningún detector
(Heath et al. 1996; Heath et al. 1997; Heath et al. 1998)	Presentar un método de evaluación basado en la opinión de personas	Cualitativa	Real sin ruido y artificial con ruido gaussiano	Bordes de objetos reales y bordes circulares de tipo “escalón”	Realce y umbralización Cruces por cero Ajuste paramétrico Supresión de no máximos locales
(Cho et al. 1997)	Presentar un método de evaluación estadística	Cuantitativa	Artificial con ruido gaussiano y real sin ruido	Borde orientado de tipo “escalón” y bordes de escenas reales	Realce y umbralización
(Shin et al. 1998a)	Presentar un método de evaluación basado en una tarea de visión.	Cuantitativa	Real con mapa de bordes definido manualmente	Bordes reales	Supresión de no máximos locales
(Dougherty y Bowyer 1998)	Proponer un sistema para la evaluación de los detectores de bordes basado el uso de imágenes reales de referencia	Cuantitativa	Real con mapa de bordes definido manualmente	Bordes de escenas reales	Realce y umbralización Supresión de no máximos locales

Tabla 4.6: Estudios comparativos de algoritmos de detección de bordes en imágenes en color (primera parte).

Autor	Propósito	Tipo de evaluación	Tipo de imagen	Tipo de borde	Tipo de detector
(Novak y Shafer 1987)	Evaluar el rendimiento de versiones en color del algoritmo de Canny (1.986).	Cualitativa.	Real.	Bordes reales.	Versiones en color del algoritmo de Canny (1.986).
(Allen y Hunt-berger 1989)	Evaluar la capacidad de detección.	Cuantitativa.	Artificial.	Bordes horizontales y verticales de tipo “escalón” provocados por cambios de color o de intensidad.	Ajuste paramétrico Cruces por cero Derivados de orientados a regiones.
	Analizar los bordes detectados e identificar la componente de color más significativa.	Cualitativa.	Real.	Bordes reales.	

Tabla 4.7: Estudios comparativos de algoritmos de detección de bordes en imágenes en color (segunda parte).

Autor	Propósito	Tipo de evaluación	Tipo de imagen	Tipo de borde	Tipo de detector
(Gauch y Hsia 1992)	Evaluar la influencia de diferentes espacios de color sobre algunos algoritmos de segmentación.	Cuantitativa y cualitativa.	Artificiales y reales con ruido gaussiano.	Bordes provocados por líneas, esquinas, uniones y esferas de diferentes colores.	Detector de bordes basado en los cruces por cero de las derivadas direccionales de segundo orden y métodos de segmentación orientados a regiones.
(Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996)	Evaluar la capacidad de detección. Comparar un detector basado en el ordenamiento de vectores con otros detectores de bordes en color.	Cuantitativa.	Artificial con ruido.	Bordes horizontales, verticales, y diagonales provocados por cambios en todas o en algunas de las componentes de color.	Detectores basados en el ordenamiento de vectores, en la suma de gradientes, en el vector gradiente o en la entropía.
		Cualitativa.	Real con ruido.	Bordes reales.	

Tabla 4.8: Estudios comparativos de algoritmos de detección de bordes en imágenes en color (tercera parte).

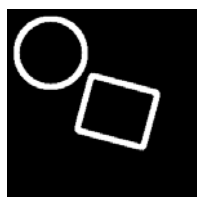
Autor	Propósito	Tipo de evaluación	Tipo de imagen	Tipo de borde	Tipo de detector
(Koschan 1995)	Comparar algoritmos de detección de bordes en color y variantes mono-cromáticas.	Cualitativa.	Artificial con ruido gaussiano.	Bordes provocados por líneas, esquinas, uniones y esferas de diferentes colores.	Realce y umbralización
			Real.	Bordes reales provocados por escenas del “mundo de los bloques”.	Supresión de no máximos locales.
(Zhu et al. 1999)	Evaluar la capacidad de detección y la sensibilidad al ruido.	Cuantitativa.	Artificial.	Bordes verticales, horizontales y diagonales de tipo “escalón” o “ranpa” provocados por variaciones de los planos de color o de intensidad.	Ordenamiento de vectores
	Valorar la calidad de los bordes.	Cualitativa.	Real con ruido mixto.	Bordes reales.	Diferencias de vectores

Capítulo 5

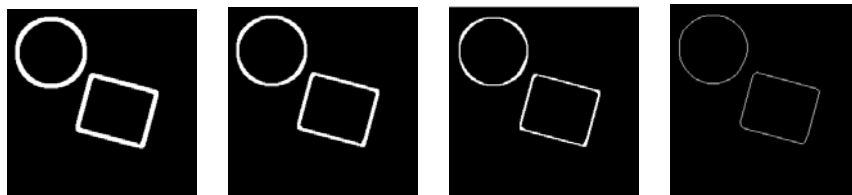
Algoritmos de adelgazamiento de bordes

5.1 Introducción

Los algoritmos de detección de bordes generan imágenes binarias con los contornos de los objetos. Desafortunadamente, los contornos generados suelen ser demasiado gruesos, y por ello es necesario aplicar un algoritmo de adelgazamiento de bordes que facilite las fases posteriores de descripción y reconocimiento de los objetos.



(a)



(b)

(c)

(d)

(e)

Figura 5.1: (a) Contornos de dos objetos y (b-e) pasos de un algoritmo de adelgazamiento de bordes.

Los algoritmos de adelgazamiento de bordes procesan el contorno de un objeto para generar otro contorno que esté compuesto sólo por líneas conexas de un punto de anchura (figura 5.1).

Los algoritmos de adelgazamiento de bordes también se denominan algoritmos de esqueletización de regiones, puesto que permiten generar el esqueleto central de los objetos.

Los algoritmos de adelgazamiento de bordes deben respetar dos condiciones: (O’Gorman 1990; Pitas 1993).

- No se pueden producir nuevas discontinuidades en los contornos: sólo se podrá eliminar un punto si se asegura que se mantiene la conectividad entre los demás puntos del contorno (figura 5.2(a)).
- No deben eliminarse puntos que formen parte del extremo final de una línea del contorno (figura 5.2(b)).

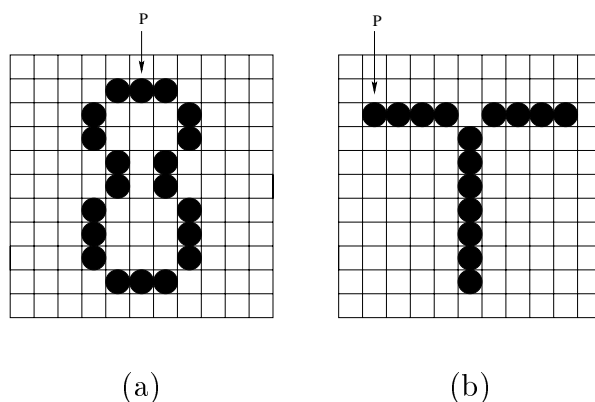


Figura 5.2: Condiciones de los algoritmos de adelgazamiento de bordes: punto P que no puede borrarse debido a que (a) provocaría una discontinuidad o (b) acortaría el extremo de una línea.

Se puede utilizar dos tipos de conectividad:

- Conectividad de orden 4: los puntos de la imagen que poseen una conectividad de orden 4 con un punto P son sus cuatro inmediatos vecinos horizontales y verticales, denominados $V_4(P)$ (figura 5.3(a)).
- Conectividad de orden 8: los puntos de la imagen que poseen una conectividad de orden 8 con un punto P son los ocho vecinos que lo rodean. En realidad, la vecindad de orden 8 no es más que la unión de los puntos integrantes de la vecindad de orden 4 y los puntos situados en las posiciones diagonales (figura 5.3(b)).

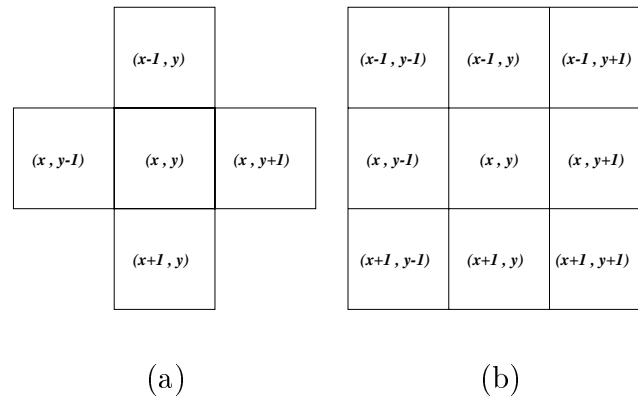


Figura 5.3: Vecindades de orden (a) 4 y (b) 8.

El tipo de conectividad influye en el funcionamiento y el resultado de los algoritmos de adelgazamiento de bordes.

La mayoría de los algoritmos de adelgazamiento son iterativos, eliminando en cada iteración la capa más exterior de un punto de anchura. Sin embargo, también existen algoritmos no iterativos que realizan la operación de adelgazamiento en un número predeterminado de pasos, independientemente del grosor del contorno de los objetos.

Se ha de indicar que los algoritmos de adelgazamiento iterativos se denominan “secuenciales” o “paralelos” dependiendo de cómo se efectúe el procesamiento de los puntos:

- Algoritmo secuencial: el resultado de procesar un punto depende de los valores de los puntos tratados previamente en la misma iteración.
- Algoritmo paralelo: el resultado de procesar un punto depende de los valores de los puntos obtenidos en la iteración anterior. Debido a ello, todos los puntos podrían ser evaluados simultáneamente o en “paralelo”.

Son muchos los algoritmos de adelgazamiento de bordes o esqueletización que han sido propuestos (Chen y Tsai 1990; Chen y Yu 1996; Davies y Plummer 1981; Gong y Bertrand 1990; Grau 1997; Krishnapuram y Chen 1990; Lam et al. 1992; Mukherjee y Chatterji 1989; O’Gorman 1990; Pal 1989; Smith 1995; Tamura et al. 1978; Wang y Zhang 1989; Wright et al. 1994). Una recopilación de las metodologías utilizadas en estos algoritmos se puede encontrar en (Lam et al. 1992). La paralelización de estos algoritmos también ha sido estudiada (Chen y Tsai 1990; Krishnapuram y Chen 1993; Sossa 1989; Zhang y Suen 1984).

A continuación se van a describir algunos de los muchos algoritmos propuestos. Esta revisión no pretende, en modo alguno, ser exhaustiva; sólo trata de mostrar algunos de los algoritmos más interesantes.

5.2 Transformación del eje “medial”

La transformación del eje “medial” (*medial axil transformation*) (Blum 1967; Escalera 2001; González y Woods 1996; Parker 1997; Sonka et al. 1993) es un algoritmo que permite obtener el esqueleto de un objeto en una iteración. Su funcionamiento es el siguiente:

- Se calcula la distancia mínima de cada punto a un punto del borde del objeto.
- Si más de un punto de borde se encuentra a esa distancia mínima del punto analizado, entonces ese punto pertenece al eje medial.

Algunos de los inconvenientes de este método son:

- Su alto coste computacional.
- El esqueleto obtenido no siempre tiene la forma del objeto (figura 5.4(a)).
- Es muy sensible al ruido (figura 5.4(b)).
- Se obtienen esqueletos diferentes dependiendo del tipo de distancia que se utilice (Parker 1997).

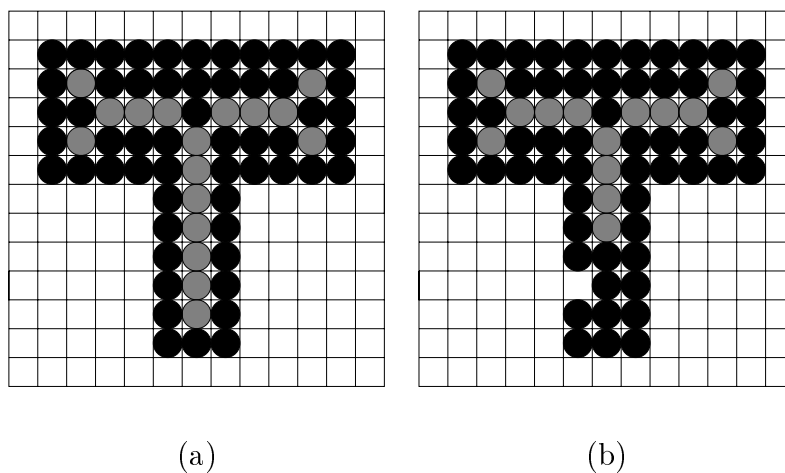


Figura 5.4: Esqueletos obtenidos por la transformación del eje “medial”: (a) imagen original; (b) imagen con ruido.

5.3 Algoritmo de Rosenfeld y Kak

El algoritmo de adelgazamiento de Rosenfeld y Kak (1.982) (Krishnapuram y Chen 1990; Krishnapuram y Chen 1993) clasifica a cada punto del contorno como:

- “Aislado” si no tiene ningún vecino marcado¹ en su vecindario.
- “Final” si tiene exactamente un vecino marcado en su vecindario.
- “8-simple” si sus vecinos marcados están conectados entre sí mediante conectividad 8.
- “4-simple” si sus vecinos marcados están conectados entre sí mediante conectividad 4.

Por tanto, se tiene que:

1. Un punto aislado no es simple.
2. Un punto final es automáticamente simple.
3. Si se borra un punto simple no se rompe la conectividad de los puntos del vecindario.

Este algoritmo establece dos restricciones para poder eliminar un punto durante el adelgazamiento:

1. No se puede borrar ningún punto final.
2. No se puede romper la conectividad.

Debido a estas restricciones, el algoritmo trata de borrar todos los puntos simples que no son puntos finales.

El algoritmo consta de cuatro fases (norte, sur, este y oeste) que se aplican de forma consecutiva en cada iteración y concluye cuando en una iteración no se puede eliminar ningún punto.

P_7	P_0	P_1
P_6	P	P_2
P_5	P_4	P_3

Figura 5.5: Vecindario de un punto P .

Teniendo en cuenta la numeración de la figura 5.5, un punto marcado P es borrado si se verifican las condiciones correspondientes a cada fase:

¹Se va a indicar que un punto está “marcado” si pertenece al contorno y como “no marcado” en caso contrario.

- Norte:

$$P_0 = 0 \wedge B_m(P) > 1 \wedge N_m(P) = 0 \quad (5.1)$$

- Sur

$$P_4 = 0 \wedge B_m(P) > 1 \wedge S_m(P) = 0 \quad (5.2)$$

- Este

$$P_2 = 0 \wedge B_m(P) > 1 \wedge E_m(P) = 0 \quad (5.3)$$

- Oeste

$$P_6 = 0 \wedge B_m(P) > 1 \wedge O_m(P) = 0 \quad (5.4)$$

donde

- m representa la conectividad: 4 u 8.
- $B_m(P)$ es el número de vecinos marcados de P usando la m -conectividad y es utilizado para determinar si un punto es “final”:

– $B_8(P)$ se define como

$$B_8(P) = \sum_{k=0}^7 P_k \quad (5.5)$$

– y $B_4(P)$ como

$$B_4(P) = \sum_{k=0}^3 P_{2k} \quad (5.6)$$

- $N_m(P)$ permite identificar si un punto en sentido norte² es “ m -simple”:

– $N_4(P)$ se define como

$$N_4(P) = P_2 \overline{P}_4 P_6 + P_4 \overline{P}_5 P_6 + P_2 \overline{P}_3 P_4 \quad (5.7)$$

– y $N_8(P)$ como

$$N_8(P) = P_2 \overline{P}_4 P_6 + \sum_{k=0}^3 \overline{P}_{2k} P_{2k+1} \overline{P}_{2k+2} \quad (5.8)$$

donde

- P_k representa el valor correspondiente al punto k -ésimo del vecindario de P (figura 5.5).
- $P_k = 1$ si el punto está marcado y $P_k = 0$ en caso contrario.
- $\overline{P}_k = 1 - P_k$.
- $P_k = P_{k \bmod 8}$ si $k \geq 8$.

Los valores de $S_m(P)$, $E(P)$ y $O(P)$ se definen de forma similar.

²Se dice que un punto está en sentido norte si su vecino P_0 no está marcado.

5.4 Algoritmo de Zhang y Suen

El algoritmo de Zhang y Suen (1984) permite obtener el esqueleto de un objeto o el contorno adelgazado mediante la aplicación de iteraciones compuestas por dos fases:

1. Primera fase

$$2 \leq B_8(P) \leq 6 \tag{5.9}$$

$$A(P) = 1 \tag{5.10}$$

$$P_0 \ P_2 \ P_4 = 0 \tag{5.11}$$

$$P_2 \ P_4 \ P_6 = 0 \tag{5.12}$$

2. Segunda fase

$$2 \leq B_8(P) \leq 6 \tag{5.13}$$

$$A(P) = 1 \tag{5.14}$$

$$P_0 \ P_2 \ P_6 = 0 \tag{5.15}$$

$$P_2 \ P_6 \ P_8 = 0 \tag{5.16}$$

donde

- $A(P)$ es el número de transiciones de puntos no marcados a puntos marcados (transiciones “0 1”) en el conjunto ordenado $P_0, P_1, \dots P_7$,
- y $B_8(P)$ es el número de vecinos marcados de P (ecuación 5.5).

El algoritmo de Zhang y Suen no permite obtener siempre un esqueleto o contorno unitario, es decir, compuesto por líneas de un punto de grosor. Considérese, por ejemplo, la situación que se indica en la figura 5.6 (Tamura et al. 1978; Sossa 1989). El punto P no puede ser borrado por el algoritmo de Zhang y Suen porque no se verifica que $A(P) = 1$.

1	0	0	0
1	0	0	0
1	P	0	0
0	1	1	1

Figura 5.6: Ejemplo en el que el algoritmo de Zhang y Suen no genera un esqueleto o contorno unitario.

Juan Humberto Sossa (1.990) propuso una variante del algoritmo de Zhang y Suen (1.984) que sí permite obtener un contorno o esqueleto unitario. Dicha variante consta de los siguientes pasos:

1. Aplicar el algoritmo de Zhang y Suen.
2. Recorrer horizontalmente la imagen generada en el paso anterior hasta encontrar un punto marcado P que sea horizontal o verticalmente adyacente a otro punto marcado y que también esté horizontal o verticalmente aislado de otros puntos marcados. Si éste es el caso, el punto P puede ser borrado si se cumple alguna de las siguientes condiciones (figuras 5.5 y 5.7):

$$\overline{P}_7 \ P_2 \ P_4 = 1, \quad (5.17)$$

$$\overline{P}_1 \ P_4 \ P_6 = 1, \quad (5.18)$$

$$\overline{P}_3 \ P_6 \ P_0 = 1, \quad (5.19)$$

$$\overline{P}_5 \ P_0 \ P_2 = 1 \quad (5.20)$$

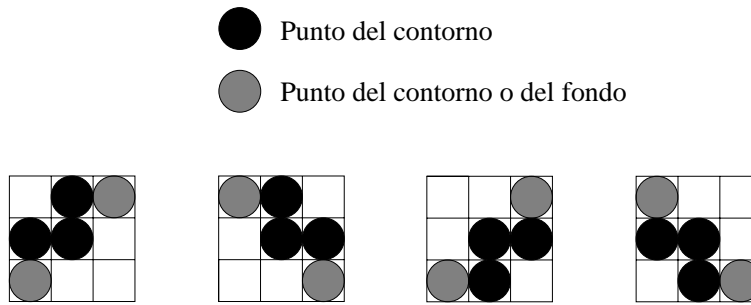


Figura 5.7: Casos en los que el algoritmo de Sossa permite borrar el punto central.

5.5 Algoritmo de Wang y Zhang

El algoritmo de Wang y Zhang (1.989) es un método iterativo compuesto por dos fases en las que se eliminan los puntos marcados que cumplan las siguientes condiciones (figura 5.5):

1. Fase sudeste:

$$1 < B_8(P) < 7 \quad (5.21)$$

$$A(P) = 1 \vee C(P) = 1 \quad (5.22)$$

$$P_2 + P_4 \ P_0 \ P_6 = 0 \quad (5.23)$$

2. Fase noroeste:

$$1 < B_8(P) < 7 \quad (5.24)$$

$$A(P) = 1 \vee C(P) = 1 \quad (5.25)$$

$$P_0 + P_6 \ P_2 \ P_4 = 0 \quad (5.26)$$

donde

- $B_8(P)$ es el número de vecinos marcados de P (ecuación 5.5),
- $A(P)$ es el número de transiciones de puntos no marcados a puntos marcados cuando se recorre el vecindario de P en el sentido de las agujas del reloj
- y $C(p)$ se define como

$$C(P) = \begin{cases} 1 & \text{Si } P_{k-1} + P_k + P_{k+1} + P_{k+4} = 0 \\ & \wedge P_{k+3} = 1 \wedge P_{k+5} = 1 \\ & \wedge (k = 1 \vee k = 3) \\ 0 & \text{En cualquier otro caso} \end{cases} \quad (5.27)$$

Krishnapuram y Chen (1.990,1993) propusieron una modificación del algoritmo de Wang-Zhang que consiste en no borrar el punto P cuando su vecindario coincida con alguno de los patrones que aparecen en la figura 5.8. Con esta modificación se pretende conseguir un esqueleto o contorno adelgazado de mejor calidad.

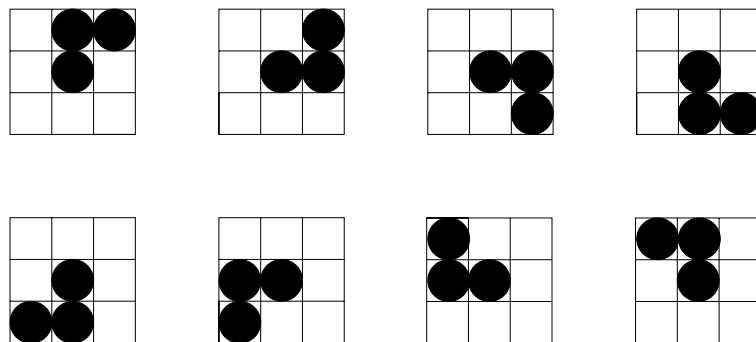


Figura 5.8: Modificación del algoritmo de Wang-Zhang: casos en los que no se debe borrar el punto central.

Krishnapuram y Chen (1.990,1993) comprobaron que las redes neuronales artificiales podían ser aplicadas a los algoritmos de adelgazamiento de contornos y esqueletización de regiones. Los algoritmos utilizados en sus trabajos fueron los de Rosenfeld y Kak (1.982) (versiones basadas en la conectividad de orden 8 y orden 4) y Wang y Zhang (1.989) (versiones original y modificada).

Posteriormente (Gallego-Ayllón 1998; Fernández-García et al. 1999), se utilizaron redes neuronales artificiales para diseñar versiones de los algoritmos de Rosenfeld y Kak y de Wang y Zhang que podían ser ejecutadas utilizando 2 o más procesadores, es decir, se obtuvieron codificaciones “paralelas” de los algoritmos.

5.6 Algoritmo de la imagen densa

El algoritmo propuesto por Chen y Tsai (1.990) utiliza una imagen auxiliar, denominada “imagen densa”, para obtener el esqueleto de la región o el contorno adelgazado.

Las dimensiones de la imagen densa son menores en una unidad que las dimensiones de la imagen original. El valor de cada punto de la imagen densa se obtiene a partir de la combinación de los valores de cuatro puntos vecinos de la imagen original (figura 5.9). Solamente cuando estos cuatro puntos de la imagen de entrada tienen el valor “1” (punto marcado), el punto de la imagen densa tendrá el valor “1”, en cualquier otro caso dicho punto tendrá como valor “0”.

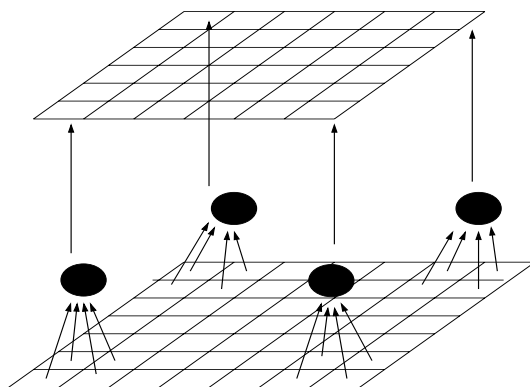


Figura 5.9: Esquema de la obtención de la imagen densa.

Puesto que puede ser muy costoso mantener dos imágenes simultáneamente (la imagen de entrada y la imagen densa), sólo es necesario generar los valores de los puntos de la imagen densa que se correspondan con los puntos que se estén procesando en cada instante.

La aplicación del algoritmo consta de las siguientes fases:

1. Eliminación de los puntos cuyos puntos vecinos en una ventana de tamaño 5×3 no coincidan con ninguna máscara del conjunto $a1 - a10$ (figura 5.10) y cuyos puntos vecinos en una ventana de tamaño 4×4 de la imagen densa no coincidan con el conjunto de máscaras $b1 - b9$ (figura 5.11).
2. Eliminación de todos los puntos cuyo vecindario coincida con alguna de las máscaras del conjunto $a11 - a18$ (figura 5.14). Gracias a esta acción se suprime el ruido que se haya generado durante el paso anterior.
3. Restauración³ de todos los puntos cuyo vecindario coincidan con alguna de las máscaras $a19 - a26$ (figura 5.15).

³Este proceso consistirá en poner el valor ‘1’ en el punto seleccionado.

Este conjunto de pasos se repetirá sobre la imagen hasta que ningún punto de la imagen sea eliminado.

Los símbolos de las máscaras del algoritmo de la imagen densa poseen el siguiente significado:

- Un punto negro indica que en dicha situación existe un punto marcado (punto con valor 1).
- Un punto blanco indica que en dicha situación existe un punto no marcado (punto con valor 0).
- Un cuadrado indica que al menos uno de los puntos donde se encuentra (siempre se encuentra en más de un punto) debe tener como valor 0, es decir, no estar marcado.

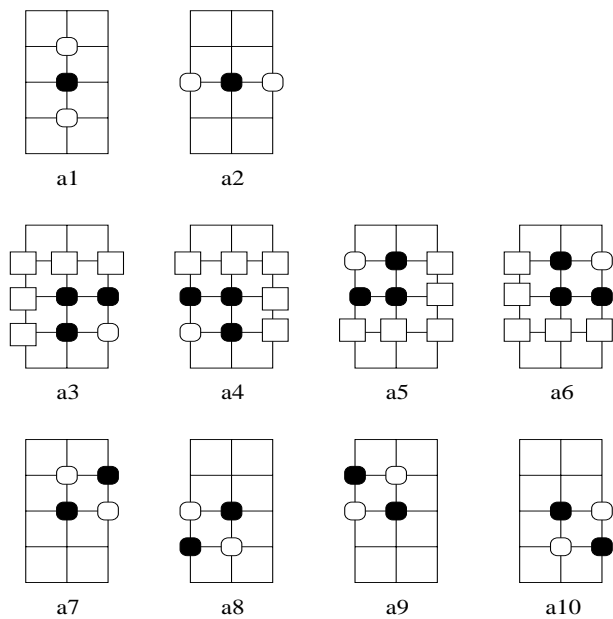


Figura 5.10: Máscaras a1 - a10.

Las primeras máscaras básicas para la aplicación del algoritmo se pueden observar en la figura 5.10. El propósito de estas primeras máscaras, denominadas “conjunto de máscaras a1 – a10”, consiste en evitar que determinados puntos sean borrados:

- Las máscaras a1 y a2 son usadas para mantener los puntos de las líneas horizontales y verticales, respectivamente.
- Las máscaras a3 – a6 se usan para preservar los puntos de las esquinas del objeto.

- Por último, las máscaras $a7 - a10$ se usan para mantener las líneas diagonales.

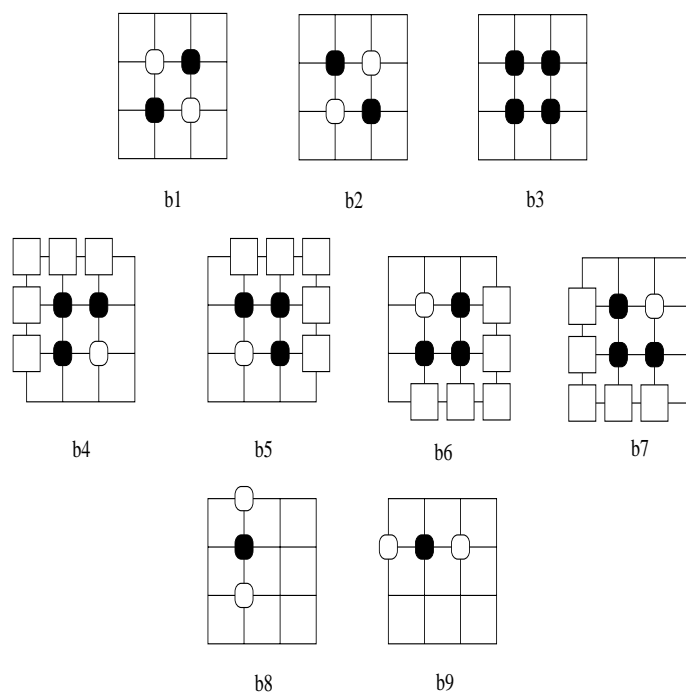


Figura 5.11: Máscaras b1-b9.

Al aplicar el conjunto de máscaras $a1 - a10$ sobre la imagen, la mayoría de los puntos son borrados, incluyendo los puntos interiores. Para evitar este problema, se utilizan las máscaras $b1 - b9$ que se aplican sobre la imagen densa (figura 5.11). El propósito de cada una de estas máscaras es el siguiente:

1. Las máscaras $b1$ y $b2$ son aplicadas para preservar las líneas diagonales.
2. La máscara $b3$ es usada para preservar áreas gruesas.
3. El conjunto de máscaras $b4 - b7$ se usa para mantener los puntos de las esquinas.
4. Las dos últimas máscaras $b8$ y $b9$ son aplicadas para preservar, respectivamente, las líneas horizontales y verticales.

La relación que se establece entre las máscaras⁴ de tipo A y las máscaras de tipo B se puede observar en la figura 5.12.

⁴Las máscaras que se aplican sobre la imagen de entrada serán denominadas “máscaras de tipo A”, mientras que las máscaras que se aplican sobre la imagen densa serán “máscaras de tipo B”.

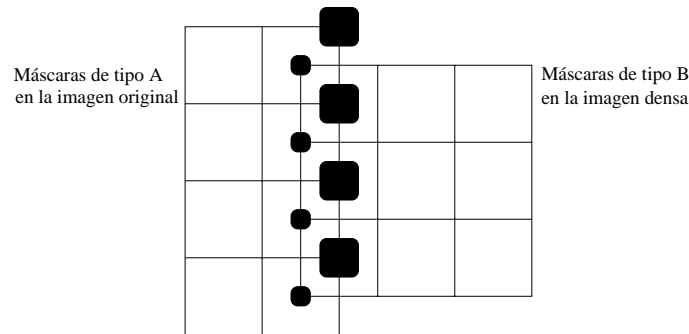


Figura 5.12: Relación entre las máscaras de tipo A y de tipo B.

El conjunto de máscaras $a1$ - $a10$ y las máscaras de tipo B están diseñadas para que puedan ser aplicadas de forma simultánea, ya que operan sobre imágenes distintas. Sin embargo, el resultado de aplicar este conjunto de máscaras produce básicamente dos problemas:

- Las máscaras no se pueden usar para eliminar los puntos aislados. Como consecuencia de esto, aparecerá en el esqueleto diversos puntos que lo deformarán.
- El uso de los conjuntos de máscaras $a1 - a10$ y $b1 - b9$ provoca el borrado de determinados puntos que están situados en el centro de una *unión T*.

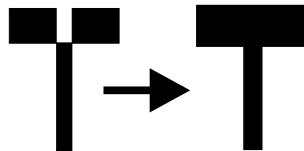


Figura 5.13: Restauración de una unión T.

Para resolver estos dos problemas se usan dos conjuntos de máscaras que se aplicarán sobre la imagen de entrada justo después de aplicar las máscaras $a1 - a10$ y $b1 - b9$. Estos conjuntos de máscaras se denominan, respectivamente, $a11 - a18$ y $a19 - a26$ y se clasifican como máscaras de tipo A, ya que se aplican sobre la imagen de entrada. El cometido de cada conjunto es el siguiente:

- El conjunto de máscaras $a11$ - $a18$ (figura 5.14) se utiliza para eliminar todos aquellos puntos que forman el ruido.
- El conjunto de máscaras $a19$ - $a26$ (figura 5.15) se utiliza para restaurar, los puntos borrados que formaban parte de una unión *T*. Se puede observar un ejemplo del procedimiento de restauración en la figura 5.13.

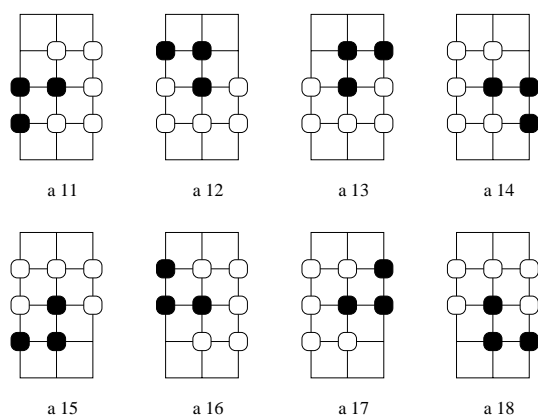


Figura 5.14: Máscaras a11-a18.

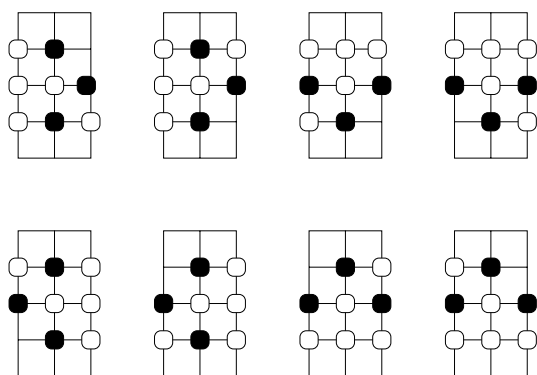


Figura 5.15: Máscaras a19-a26.

5.7 Algoritmo de adelgazamiento “ $k \times k$ ”

Lawrence O’Gorman (1.990) propuso un algoritmo de adelgazamiento de bordes denominado “adelgazamiento $k \times k$ ” ($k \times k$ *thinning*) que posee dos versiones:

- Secuencial: el resultado de procesar un punto depende del resultado obtenido en los puntos tratados previamente en la misma iteración.
- Paralela: los puntos son procesados independientemente.

El adelgazamiento $k \times k$ consiste básicamente en examinar regiones de la imagen de tamaño $k \times k$, de modo que, atendiendo a unos determinados criterios, los $(k - 2) \times (k - 2)$ puntos que no forman parte de la periferia de la región son eliminados o no.

El valor de k tiene gran importancia puesto que influye en el número de iteraciones necesarias para llevar a cabo el adelgazamiento y en la calidad del resultado. A medida que aumenta el valor de k , disminuye el número de itera-

ciones, mientras que la calidad del resultado empeora, produciéndose pequeñas deformaciones.

Los requisitos de este algoritmo son los siguientes:

1. El contorno de cada objeto debe ser adelgazado hasta quedar reducido a líneas con la menor anchura posible pero manteniendo la conectividad.
2. El resultado obtenido tras el adelgazamiento debe mantener como mínimo una conectividad de orden 8 entre los puntos.
3. Se deben mantener aquellos puntos situados en los extremos finales de las líneas. Este requisito depende fuertemente del valor de k , puesto que, cuando el valor de k es grande, el número de puntos eliminados por cada operación es mayor, con lo que existe una mayor probabilidad de que alguno de ellos sea parte de un extremo.
4. La silueta final obtenida debe aproximarse al esqueleto central del contorno original. Al igual que en el requisito anterior, la aproximación depende en gran medida del valor de k , obteniéndose aproximaciones más precisas con valores pequeños.
5. Debe minimizarse el número de “picos” o “espolones” extraños. Cuando k es grande, la silueta obtenida suele presentar pequeñas prolongaciones más o menos perpendiculares al contorno, como las de la figura 5.16.

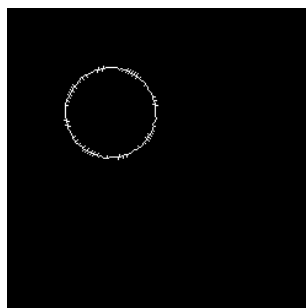


Figura 5.16: Contorno de un objeto que presenta “espolones” o “picos”.

Este algoritmo utiliza una ventana de $k \times k$ puntos que posee las siguientes características (figura 5.17):

- Una ventana $V(x, y, k)$ es una región de la imagen de tamaño $k \times k$ y centrada⁵ en (x, y) , cuyos puntos tienen unas coordenadas $(x + r, y + r)$ que

⁵Cuando el valor de k es par hay que tener en cuenta que no existe un punto central, por lo que se deberá tomar como tal uno de los puntos más cercanos al centro teórico de la región.

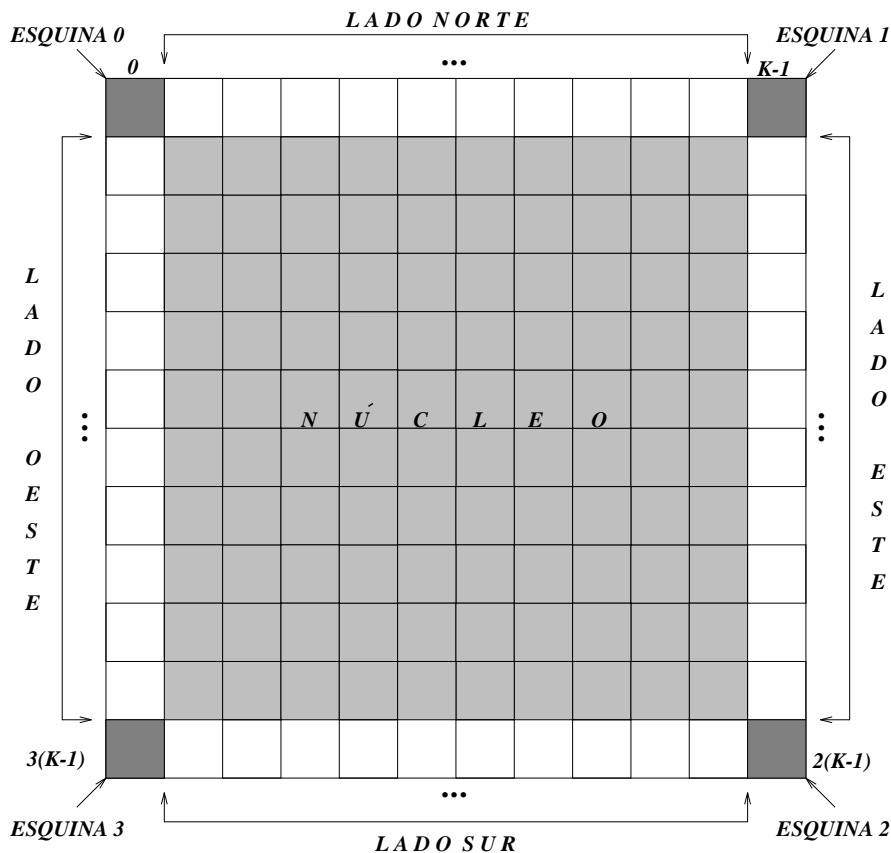


Figura 5.17: Algoritmo de O'Gorman (1.990): estructura de una ventana $V(x, y, k)$.

verifican

$$-E \left[\frac{k-1}{2} \right] \leq r \leq E \left[\frac{k}{2} \right] \quad (5.28)$$

donde $E[z]$ denota la parte entera de z .

- El núcleo $R(x, y, k)$ de $V(x, y, k)$ está formado por los $(k-2) \times (k-2)$ puntos que no forman parte de la periferia de $V(x, y, k)$. Sus puntos varían en el rango $(x+r, y+r)$, donde

$$-E \left[\frac{k-3}{2} \right] \leq r \leq E \left[\frac{k-2}{2} \right] \quad (5.29)$$

- El vecindario η de $V(x, y, k)$ está formado por los $4(k-1)$ puntos que forman la periferia de la ventana y que rodean al núcleo. Cada punto de η será referido como η_i , donde $i = 0, 1, \dots, 4(k-1)-1$, tomando η_0 como el punto situado en la esquina superior izquierda de $V(x, y, k)$, y los demás η_i serán los situados en posiciones consecutivas de la periferia, recorriendo ésta en el sentido de las agujas del reloj.

- Las esquinas son los cuatro puntos de η situados en la parte superior izquierda, superior derecha, inferior derecha e inferior izquierda de $V(x, y, k)$. Se corresponden respectivamente con $\eta_0, \eta_{k-1}, \eta_{2(k-1)}$ y $\eta_{3(k-1)}$.
- Los lados son las regiones de η comprendidas entre cada par de esquinas. Hay cuatro lados: norte (L_N), este (L_E), sur (L_S) y oeste (L_O), formados por los siguientes puntos:

$$\begin{aligned} L_N &= \{\eta_1, \dots, \eta_{k-2}\} \\ L_E &= \{\eta_k, \dots, \eta_{2(k-1)-1}\} \\ L_S &= \{\eta_{2(k-1)+1}, \dots, \eta_{3(k-1)-1}\} \\ L_O &= \{\eta_{3(k-1)+1}, \dots, \eta_{4(k-1)-1}\} \end{aligned}$$

Antes de establecer los criterios de adelgazamiento es necesario tener en cuenta las siguientes definiciones:

- Un punto ON es aquél que forma parte de un borde, mientras que uno OFF es el que forma parte del fondo de la imagen. Los puntos ON vendrán representados por el valor 1, y se mostrarán mediante el color negro, mientras que los puntos OFF tendrán el valor 0 y serán de color blanco.
- Un núcleo se denomina ON si todos sus puntos son ON , y OFF si todos sus puntos son OFF . Igualmente sucede con los lados, que pueden ser ON u OFF .
- Cuando se va a eliminar un punto, éste puede tomar un valor intermedio (BORRADO) antes de ser puesto a OFF .
- El valor de k indica el tamaño del lado de $V(x, y, k)$ (no confundir con los lados de η). Cada uno de los lados de $V(x, y, k)$ está formado por dos esquinas consecutivas y el lado de η que delimitan ambas esquinas. Además, k también indica el nivel de adelgazamiento, donde $k \geq 3$.
- $\phi_0(\eta)$ es la longitud⁶ máxima de las longitudes de todas las cadenas que se pueden formar con puntos OFF de η 4-conexos.
- $\phi_1(\eta)$ es la longitud máxima de las longitudes de todas las cadenas que se pueden formar con puntos ON de η 8-conexos.
- $\chi(\eta)$ es el número de cadenas de puntos ON 4-conexos u 8-conexos pertenecientes a η , el cual debe tener al menos un punto OFF . A $\chi(\eta)$ también se la llama conectividad del núcleo, y su valor viene dado por la siguiente expresión:

$$\chi(\eta) = T_1 + T_2 + T_3 \quad (5.30)$$

⁶Expresada en número de puntos.

donde

$$T_1 = \frac{1}{2} \sum_{i=0}^{4(k-1)-1} |\eta_i - \eta_{i-1}|, \text{ donde } n(k-1) \notin \{i, i-1\} \quad (5.31)$$

$$T_2 = \frac{1}{2} \sum_{m=n(k-1)} |\eta_{m+1} - \eta_{m-1}| \quad (5.32)$$

$$T_3 = \sum_{m=n(k-1)} \eta_m |\eta_m - \eta_{m-1}| |\eta_m - \eta_{m+1}| \quad (5.33)$$

para $n = 0, 1, 2, 3$ y $\eta_{-1} = \eta_{4k-5}$

El término T_1 cuenta las transiciones $ON \rightarrow OFF$ u $OFF \rightarrow ON$ entre dos puntos contiguos de η , de modo que ninguno de ambos puntos sea una esquina, es decir, ambos deben pertenecer a uno de los lados de η . La expresión 5.32 cuenta las transiciones $ON \rightarrow OFF$ u $OFF \rightarrow ON$ entre dos puntos de η separados por una esquina. Por último, T_3 cuenta las transiciones $OFF \rightarrow ON \rightarrow OFF$ entre tres puntos de η contiguos, siendo el segundo de ellos una esquina ON . Para k igual a 3 no tiene sentido calcular T_1 , ya que cada uno de los cuatro lados de η está formado por un punto (figura 5.18).

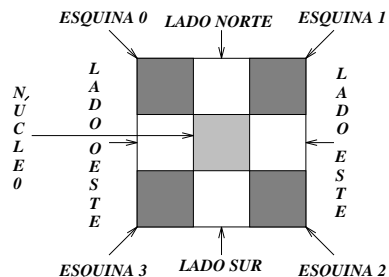


Figura 5.18: Partes de la ventana de 3×3 puntos del algoritmo de O'Gorman.

Teniendo en cuenta todas las definiciones anteriores, para una ventana $V(x, y, k)$, si su núcleo $R(x, y, k)$ es ON , entonces este último puede ser eliminado (no en todos los casos) si cumple las siguientes condiciones:

1. $\chi(\eta) = 1$
2. $\phi_1(\eta) > k - 2$
3. $\phi_0(\eta) > k - 2$

El primer criterio asegura que se mantenga la conectividad del contorno de los objetos. Cuando $\chi(\eta) = 1$, en η puede formarse una cadena no cerrada de puntos ON 4-conexos u 8-conexos, de tal modo que la eliminación del núcleo no

rompe la conectividad de cualquier cadena de η de cualquier longitud formada por puntos ON. En el caso de que $\chi(\eta) > 1$, entonces hay más de una cadena de puntos ON conexos. Si se eliminara el núcleo, se produciría una separación entre dichas cadenas, rompiéndose la conectividad. Por último, si $\chi(\eta) = 0$, entonces η está formado completamente por puntos ON o por puntos OFF.

En cuanto al segundo criterio, su objetivo es mantener los extremos de las líneas. Para un nivel de adelgazamiento k , sólo se consideran extremos de una línea a aquellos cuya anchura es menor o igual que la longitud de un lado de η , es decir, $k - 2$. Si un núcleo está rodeado por un vecindario que contiene una cadena de longitud menor o igual que $k - 2$ puntos ON y conexos, entonces dicho núcleo constituye un extremo de una línea en el nivel k de adelgazamiento, y, por tanto, no es eliminado. Cuando $\chi(\eta) = 1$, $\phi_1(\eta)$ indica el número de puntos ON del vecindario η .

Por último, el tercer criterio establece una condición opuesta al criterio anterior. Aquí se pretende evitar la erosión interna del contorno, ya que si no se provocarían “huecos” en la silueta del objeto.

5.7.1 Versión secuencial

La versión secuencial del algoritmo O’Gorman (1.990) trata los puntos de la imagen en un orden secuencial, de forma que se examina la imagen de arriba hacia abajo y de izquierda a derecha.

Este algoritmo también se denomina “multivalor”, surgiendo este nombre del hecho de que, además de los dos valores binarios originales de la imagen, se introducen otros dos valores para mantener información auxiliar que permita decidir qué puntos son susceptibles de ser eliminados. Estos dos valores auxiliares son *BORRADO* y *BORRADO_A*.

El modo de operar del algoritmo es el siguiente: se parte de la imagen original binaria con puntos ON y OFF. Dado el valor k , se desplaza secuencialmente la ventana $V(x, y, k)$ por la imagen en el orden mencionado anteriormente. En cada posición, se examinan los puntos que forman parte del núcleo.

Si el núcleo no es ON, la ventana es desplazada a la siguiente posición. Si, por el contrario, el núcleo es ON, entonces se comprueba si se verifican los tres criterios de adelgazamiento. Si no se cumplen los tres criterios, se reduce el tamaño de la ventana en una unidad, pasando a una ventana $V(x, y, k - 1)$. En esta ventana se vuelven a comprobar los criterios. Si siguen sin cumplirse, entonces se vuelve a reducir el tamaño de la ventana sucesivamente en una unidad hasta conseguir un tamaño de ventana⁷ $k' \times k'$ en el que se cumplan los tres criterios, en cuyo caso se hará una segunda verificación (explicada más adelante),

⁷El tamaño mínimo permitido de la ventana es de 3×3 .

y si es válida, entonces todos los puntos del núcleo tomarán un valor *BORRADO* o *BORRADO_A*, los cuales serán explicados posteriormente.

Si no es posible encontrar un valor k' que permita cumplir las verificaciones, los puntos del núcleo mantienen su valor original y se desplaza la ventana $V(x, y, k)$ a la siguiente posición.

Hay que aclarar que a la hora de verificar los criterios de adelgazamiento, todos aquellos puntos del vecindario de la ventana de tamaño $k' \times k'$ (para $k' = k, k - 1, \dots, 3$) que tengan un valor *BORRADO* o *BORRADO_A* (valores que habrán adquirido en operaciones previas en la misma iteración) serán considerados como *ON*. Si se obtiene ese valor k' que permita cumplir los tres criterios, entonces se hace una segunda verificación más compleja en la que, ahora sí, se tienen en cuenta los valores *BORRADO* y *BORRADO_A*.

La primera verificación se hace para determinar si el núcleo forma parte del contorno de un objeto, y también para comprobar que dicho núcleo no es necesario para mantener la conectividad o un extremo de una línea.

La segunda verificación se hace para asegurar que la eliminación del núcleo junto con eliminaciones previas no romperá la conectividad o eliminará por completo una región aislada. Esta segunda verificación implica examinar los cuatro lados de η (L_N, L_E, L_S y L_O) junto con las dos esquinas que los delimitan respectivamente. La forma de examinar cada lado es la siguiente: todos los puntos con valor *BORRADO* o *BORRADO_A* que pertenezcan al lado en cuestión o a sus dos esquinas adyacentes se consideran como *OFF*, mientras que los puntos de las restantes esquinas y lados con alguno de esos dos valores se consideran como *ON*. Pero hay una excepción. Si el punto situado en la esquina superior izquierda (η_0) tiene un valor *BORRADO_A*, entonces es considerado como *ON* en todas y cada una de las comprobaciones de los cuatro lados. Si se realiza lo anterior para cada lado y se comprueba el primer criterio, resultando que $\chi(\eta) = 1$ en los cuatro casos, entonces los puntos del núcleo se marcan como *BORRADO* o como *BORRADO_A*. Si no, entonces se mantienen sus valores originales y se vuelve a disminuir el valor de k' , volviendo a repetir el proceso desde el principio hasta conseguir que se cumplan ambas verificaciones, o hasta llegar a un valor $k' = 3$ para el que no se cumplan. En este último caso se desplaza la ventana $V(x, y, k)$ a la siguiente posición.

La diferenciación entre *BORRADO* y *BORRADO_A* proviene del hecho de que el segundo criterio de adelgazamiento no es suficiente para mantener los extremos de las líneas orientadas en el mismo sentido en el que se recorre la imagen, es decir, las líneas en dirección noroeste (135°). Por ello se introduce el concepto de “ancla”. Un ancla es un núcleo situado en el extremo de una línea diagonal orientada en la dirección noroeste. Este núcleo se caracteriza por formar parte de una ventana en la que L_O, L_N , así como las dos esquinas que delimitan a L_N , están formados por puntos *OFF* (figura 5.19). De este modo, si

al cumplirse las dos verificaciones el núcleo es un ancla, sus puntos tomarán un valor $BORRADO_A$. Si el núcleo no es un ancla, entonces el valor que tomarán sus puntos será $BORRADO$. Es importante aclarar que el núcleo en cuestión es el de la ventana de tamaño $k' \times k'$ y no el original perteneciente a $V(x, y, k)$.

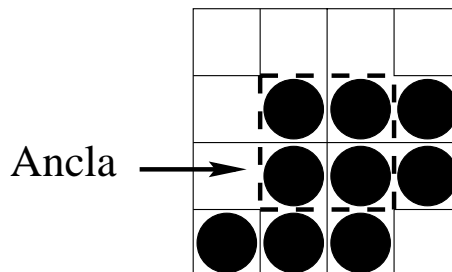


Figura 5.19: Ejemplo de un núcleo de tipo ancla para una ventana $V(x, y, 4)$.

Una vez recorrida la imagen por completo, todos los puntos con valor $BORRADO$ o $BORRADO_A$ se transforman en OFF y se repite el proceso de nuevo, comenzando una nueva iteración. La operación de adelgazamiento termina cuando en una iteración no se consigue poner ningún punto con un valor $BORRADO$ o $BORRADO_A$.

Resumiendo, el algoritmo puede ser descrito mediante la realización de los siguientes pasos:

1. Inicializar (x, y) de modo que sobre su posición quede centrada⁸ la ventana $V(x, y, k)$, situando ésta sobre la esquina superior izquierda de la imagen.
2. Centrar la ventana $V(x, y, k)$ sobre la posición (x, y) actual y a continuación:
 - (a) Hacer la siguiente inicialización: $k' \leftarrow k$.
 - (b) Comprobar que todos los puntos del núcleo de $V(x, y, k')$ tienen un valor igual a ON . Si no lo tienen, pasar a 2.f.
 - (c) En la ventana $V(x, y, k')$ considerar como ON cualquier punto del vecindario η con valor $BORRADO$ o $BORRADO_A$.
 - (d) Verificar los tres criterios de adelgazamiento.
 - (e) Si no se cumplen los tres criterios, pasar a 2.f. En caso contrario, para cada lado (L_N, L_E, L_S y L_O) de η y para las esquinas que lo delimitan, considerar como OFF cualquier punto $BORRADO$ o $BORRADO_A$ ⁹.
 - Si existe algún punto de los lados y esquinas restantes con uno cualquiera de estos dos valores, entonces dicho punto es considerado como ON .

⁸La localización inicial de (x, y) dependerá del valor de k , de modo que se descartarán aquellas localizaciones situadas en la periferia de la imagen sobre las que no se pueda centrar $V(x, y, k)$.

⁹Obviamente, al principio no habrá ningún punto con valor $BORRADO$ o $BORRADO_A$.

- Si la esquina superior izquierda de $V(x, y, k')$, es decir, η_0 , tiene un valor $BORRADO_A$, entonces se considera en todos los casos como ON .
 - A continuación se comprueba si se cumple el primer criterio de adelgazamiento, de forma que si $\chi(\eta) = 1$ para cada lado, los puntos del núcleo toman un valor $BORRADO_A$ si forman un núcleo de tipo ancla, o $BORRADO$ si es un núcleo normal, y a continuación pasamos a 2.g. Si el criterio no se cumple para alguno de los cuatro lados, entonces pasar a 2.f.
- (f) Si $k' > 3$, entonces se realiza la operación $k' \leftarrow k' - 1$ y se pasa a 2.b. Si $k' = 3$, se pasa a 2.g.
- (g) Si no se ha desplazado $V(x, y, k)$ por toda la imagen, incrementar las coordenadas (x, y) para pasar a la siguiente posición y volver a 2. En caso contrario, pasar a 3.
3. Si ningún punto ha tomado un valor $BORRADO$ o $BORRADO_A$ en la última iteración, la operación de adelgazamiento ha terminado. En caso contrario, cambiar a OFF cualquier punto $BORRADO$ o $BORRADO_A$ y volver a 1.

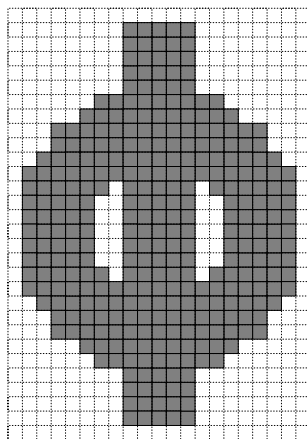
En la figura 5.20 se muestra todo el proceso de adelgazamiento 3×3 llevado a cabo sobre la silueta de un objeto. El color blanco representa los puntos con valor OFF , mientras que en orden creciente de oscuridad se muestran respectivamente los puntos con valor $BORRADO_A$ (anclas), $BORRADO$ y ON .

5.7.2 Versión paralela

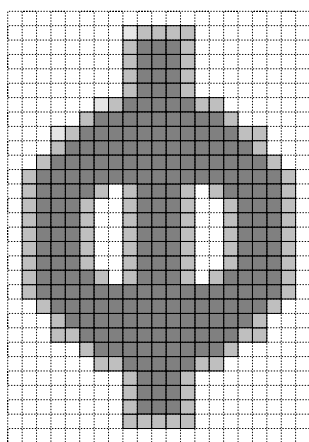
La versión paralela del algoritmo de O’Gorman (1.990) tiene la ventaja respecto a la secuencial de que pueden realizarse operaciones sobre todos los puntos de la imagen simultáneamente, siendo independientes los resultados obtenidos para un punto de los resultados obtenidos en los restantes. Sin embargo, esta ventaja está en cierto modo descompensada por el hecho de que cada iteración requiere la realización de cuatro subiteraciones o subciclos. Este algoritmo también es denominado “binario”, debido a que únicamente utiliza los valores binarios de la imagen original.

El algoritmo de adelgazamiento paralelo opera en ventanas $V(x, y, k)$ situadas sobre bordes en dirección norte, sur, este y oeste, de modo que cada subciclo afecta únicamente a una de las cuatro direcciones mencionadas. Se definen cuatro tipos de ventanas según sus direcciones (figura 5.21):

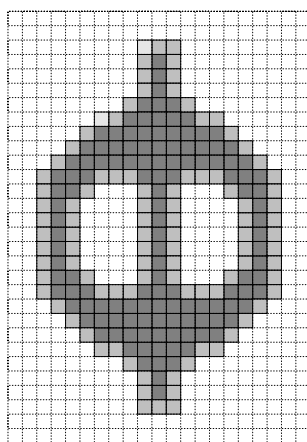
1. Una ventana en dirección norte es aquella en la que L_N está formado por puntos OFF .



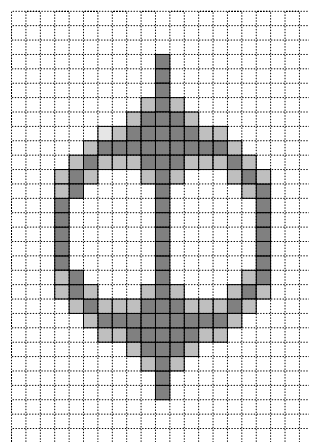
(a)



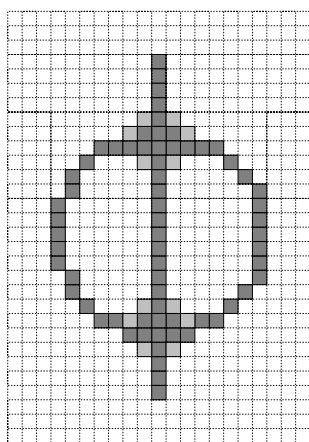
(b)



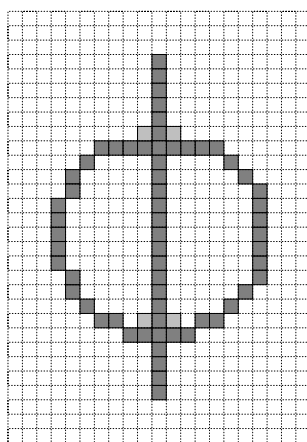
(c)



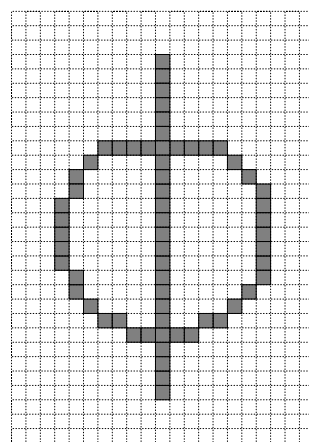
(d)



(e)



(f)



(g)

Figura 5.20: (a) Silueta de un objeto; (b - g) iteraciones de la versión secuencial del algoritmo de O'Gorman de 3×3 ((O'Gorman 1990) página. 204).

2. Una ventana en dirección sur es aquella en la que L_S está formado por puntos *OFF* y la ventana no está en dirección norte.
3. Una ventana en dirección este es aquella en la que L_E está formado por puntos *OFF* y la ventana no está en dirección norte ni sur.
4. Una ventana en dirección oeste es aquella en la que L_O está formado por puntos *OFF* y la ventana no está en dirección norte, ni sur ni este.

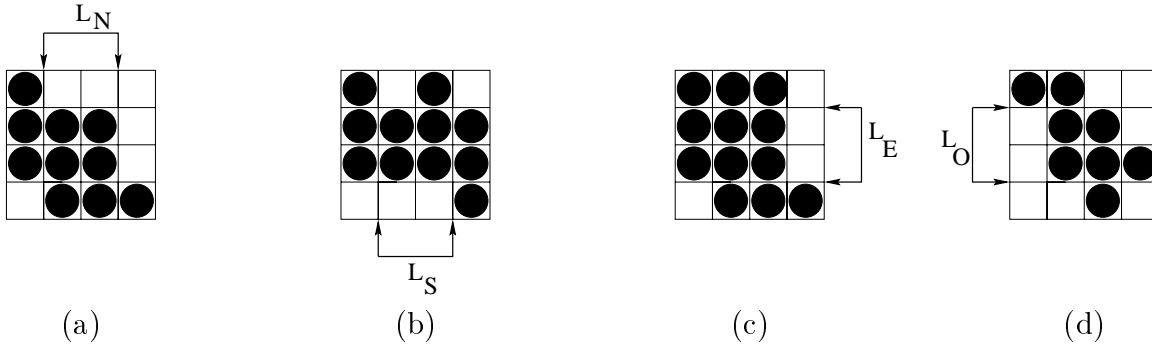


Figura 5.21: Algoritmo de O’Gorman (1.990): ejemplos de ventanas de 3×3 puntos en dirección (a) norte, (b) sur, (c) este y (d) oeste.

En cada subciclo se desplaza la ventana $V(x, y, k)$ por toda la imagen, y en la misma dirección que en el algoritmo anterior. Si tras realizar los cuatros subciclos de una iteración no se elimina ningún punto, entonces la operación de adelgazamiento habrá terminado. Los pasos del algoritmo son los siguientes:

1. $i \leftarrow 0$.
2. Determinar el punto $P = (x, y)$ de modo que sobre su posición quede centrada la ventana $V(x, y, k)$, situando ésta sobre la esquina superior izquierda de la imagen.
3. Centrar la ventana $V(x, y, k)$ sobre la posición (x, y) actual y a continuación:
 - (a) $k' \leftarrow k$.
 - (b) Verificar que la ventana está en la dirección i , donde $i \in \{0, 1, 2, 3\}$ representa la dirección norte, sur, este u oeste respectivamente. Si la verificación es incorrecta, pasar a 2.f.
 - (c) Comprobar que todos los puntos del núcleo de $V(x, y, k')$ tienen un valor igual a *ON*. Si no lo tienen, pasar a 2.f.
 - (d) Comprobar los tres criterios de adelgazamiento para $V(x, y, k')$.
 - (e) Si no se cumplen los tres criterios, pasar a 2.f. En caso contrario, eliminar los puntos del núcleo de $V(x, y, k')$, es decir, ponerlos a *OFF* y pasar a 2.g.

- (f) Si $k' > 3$, entonces se realiza la operación $k' \leftarrow k' - 1$ y volver a 2.b. Si $k' = 3$, continuar en 2.g.
 - (g) Si no se ha desplazado $V(x, y, k)$ por toda la imagen, incrementar las coordenadas (x, y) para pasar a la siguiente posición y volver a 3. En caso contrario, si $i < 3$ hacer $i \leftarrow i + 1$ y pasar a 2. Si $i = 3$, ir al paso 4.
4. Si no se ha eliminado ningún punto en ninguno de los cuatro subciclos, terminar la operación de adelgazamiento. En caso contrario, pasar a 1.

En la figura 5.22 se muestran los cuatro subciclos de la primera iteración y el resultado final de aplicar el algoritmo paralelo de adelgazamiento $k \times k$ para $k = 3$. Los puntos blancos tienen un valor *OFF*, los más oscuros son los puntos *ON*, y los que tienen un color algo más claro que estos últimos son los puntos eliminados en cada subciclo.

5.7.3 Variante para ventanas de 3×3 puntos

En el caso particular $k = 3$, se pueden hacer ciertas simplificaciones tanto en la versión secuencial como en la paralela. De acuerdo con la numeración dada a cada punto (figura 5.5), las partes de la ventana $V(x, y, 3)$ de tamaño 3×3 son (figura 5.18):

- Núcleo $R(x, y, 3)$: P .
- Vecindario η : $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$.
- Esquinas: P_1, P_3, P_5, P_7 .
- $L_N = \{p_0\}$, $L_E = \{p_2\}$, $L_S = \{p_4\}$, $L_O = \{p_6\}$.

En cuanto a los criterios de adelgazamiento, éstos quedan de la siguiente forma:

1. $\chi(\eta) = 1$
2. $\phi_1(\eta) > 1$
3. $\phi_0(\eta) > 1$

Además, el valor de $\chi(\eta)$ queda reducido a:

$$\chi(\eta) = T_2 + T_3 \quad (5.34)$$

El valor de los términos T_2 y T_3 está expresado en las ecuaciones 5.31 y 5.32.

En el algoritmo secuencial, para que un núcleo se considere como ancla debe cumplir:

- P debe ser ON .
- P_0, P_1, P_6 y P_7 deben ser OFF .

En cuanto al algoritmo paralelo, las ventanas en las cuatro direcciones deben cumplir lo siguiente:

- Ventana en dirección norte: $P_0 = OFF$.
- Ventana en dirección sur: $P_4 = OFF$ y $P_0 = ON$.
- Ventana en dirección este: $P_2 = OFF$ y $P_0, P_4 = ON$.
- Ventana en dirección oeste: $P_6 = OFF$ y $P_0, P_4, P_2 = ON$.

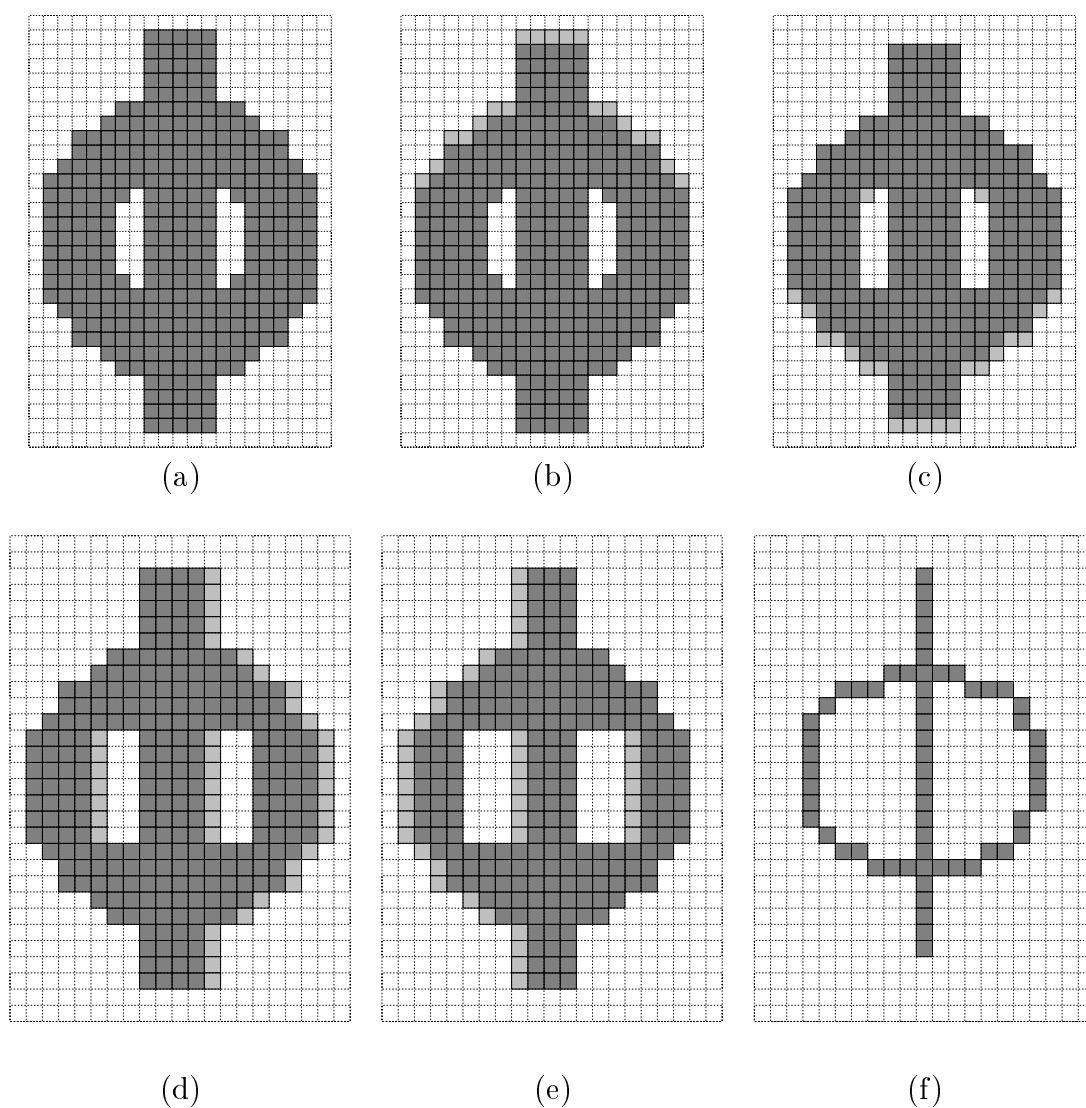


Figura 5.22: (a) Silueta de un objeto; (b - e) resultados de cada uno de los cuatro subciclos de la primera iteración de la versión paralela del algoritmo de adelgazamiento de O’Gorman de 3×3 ; (f) resultado final obtenido tras 14 subciclos. ((O’Gorman 1990) página 205).

Capítulo 6

Reconocimiento de contornos

6.1 Introducción

El reconocimiento de objetos es una de las etapas más importantes del procesamiento digital de imágenes, ya que debe identificar o clasificar un objeto en función de sus características.

Muchos de los algoritmos de reconocimiento de objetos están basados exclusivamente en el análisis de la forma (*shape analysis*), no utilizando otras características como pueden ser el color o la textura. Se puede definir la “forma” como la silueta generada por un objeto cuando es iluminado por una fuente de luz situada a una distancia infinita (Loncaric 1998).

Hay numerosas aplicaciones donde el análisis de imágenes puede ser reducido al análisis de formas: reconocimiento de caracteres, detección de tumores o lesiones, análisis de células u órganos, reconocimiento de herramientas, etc.

Los algoritmos de análisis de formas constan de tres fases:

- Representación: consiste en una identificación de los puntos que pertenecen a la forma.
- Descripción: es una caracterización de la forma que se obtiene a partir de su representación y que consta de un conjunto de propiedades o descriptores que recibe el nombre de “vector de características”
- Reconocimiento: utiliza la descripción obtenida para clasificar a un objeto.

Los métodos de análisis de formas se pueden clasificar en dos grandes grupos atendiendo al tipo de puntos del objeto que utilicen:

- Métodos basados en el contorno o métodos externos: sólo utilizan los puntos que forman parte del contorno (frontera o bordes) del objeto.
- Métodos globales o métodos internos: utilizan todos los puntos del objeto, tanto los pertenecientes al contorno como los interiores.

El análisis de formas ha sido estudiado por numerosos científicos (Aloimonos 1988; Ballard y Brown 1982; Davis 1986; Duda y Hart 1973; Haralick y Shapiro 1992; Horn 1986; Jähne 1993; Jain 1989; Levine 1985; Loncaric 1998; Nevatia 1982; van Otterloo 1991; Pavlidis 1978; Pavlidis 1980; Rosenfeld y Kak 1982b; Rosenfeld y Kak 1982a; Schalkoff 1989; Uhr 1966; Young y Fu 1986).

Puesto que el contexto en el que se sitúa el presente trabajo es el de la detección de bordes, a continuación sólo se van a describir los métodos de análisis de formas externos, es decir, métodos que sólo utilizan los puntos del contorno para la representación, descripción y reconocimiento de los objetos.

6.2 Representación de un contorno

El contorno de un objeto contenido en una imagen digital viene definido por medio de su función característica

$$f(x, y) = \begin{cases} 1 & Si (x, y) \in Contorno \\ 0 & Si (x, y) \notin Contorno \end{cases} \quad (6.1)$$

Sin embargo, para poder procesar matemáticamente el contorno, es necesario parametrizar sus puntos mediante algún sistema de representación.

Los sistemas de representación deben poseer una serie de propiedades (Mokhtarian y Mackworth 1992):

- Invariancia: si dos objetos tienen la misma forma entonces deberían también tener la misma representación.
- Unicidad: si dos objetos no tienen la misma forma, deberían tener representaciones diferentes.
- Estabilidad: si dos objetos tienen una pequeña diferencia, sus representaciones deberían tener también una pequeña diferencia y viceversa, es decir, si dos representaciones tienen alguna pequeña diferencia entonces los objetos que representan también deberían tener una pequeña diferencia.

Además de estas propiedades esenciales, un sistema de representación debería poseer las siguientes características adicionales:

- Facilidad en el cómputo de los descriptores: la representación debe facilitar el cálculo de propiedades de los contornos. Por ejemplo, si un objeto es simétrico, sería deseable poder determinar tal hecho a partir de la representación de la forma del objeto.
- Eficiencia: la representación de un objeto ha de ser computacional y espacialmente eficiente para permitir un procesamiento en tiempo real.
- Fácil de implementar: la programación y depuración del método de representación debe hacerse sin dificultad y sin tener que invertir un tiempo excesivo.

Algunas de las representaciones de contornos más utilizadas por los algoritmos de análisis de formas son:

- Representaciones basadas en los sistemas de coordenadas.
- Código de cadena.
- Signatura.
- Método de la altura de arco.
- Representaciones poligonales.
- *Splines* y *B-splines*.
- Representaciones jerárquicas.

6.2.1 Representaciones basadas en los sistemas de coordenadas

Los sistemas de coordenadas permiten diseñar fácilmente sistemas de representación de contornos. Si se utiliza la longitud de arco n como parámetro, los puntos de los contornos pueden ser representados mediante (figura 6.1):

- Coordenadas cartesianas:

$$P(n) = (x_n, y_n) \tag{6.2}$$

- Coordenadas polares respecto del eje de coordenadas:

$$P(n) = (r_n, \theta_n) \tag{6.3}$$

- Coordenadas polares respecto del centroide o centro de gravedad del objeto:

$$P(n) = (r_n, \theta_n) \quad (6.4)$$

El uso del centroide permite obtener una representación del contorno que es invariante a traslaciones. Esta parametrización también es invariante a rotaciones si se elige como primer punto al punto del contorno que esté más alejado del centroide y que, además, pertenezca al eje de mínima inercia.

Si se desea que esta parametrización sea invariante a cambios de escala, se puede normalizar el tamaño del contorno dividiendo los radios por el valor del radio máximo, consiguiéndose que los radios varíen entre 0 y 1.

- Representación tangencial: se identifica un punto en función del ángulo de la recta tangente en ese punto (Sonka et al. 1993):

$$P(n) = \theta(n) = \theta_n \quad (6.5)$$

El ángulo de la recta tangente puede ser medido respecto del eje de abscisas (Sonka et al. 1993) o respecto del ángulo tangente en un punto inicial (Zahn y Roskies 1972; Bennet y McDonald 1975).

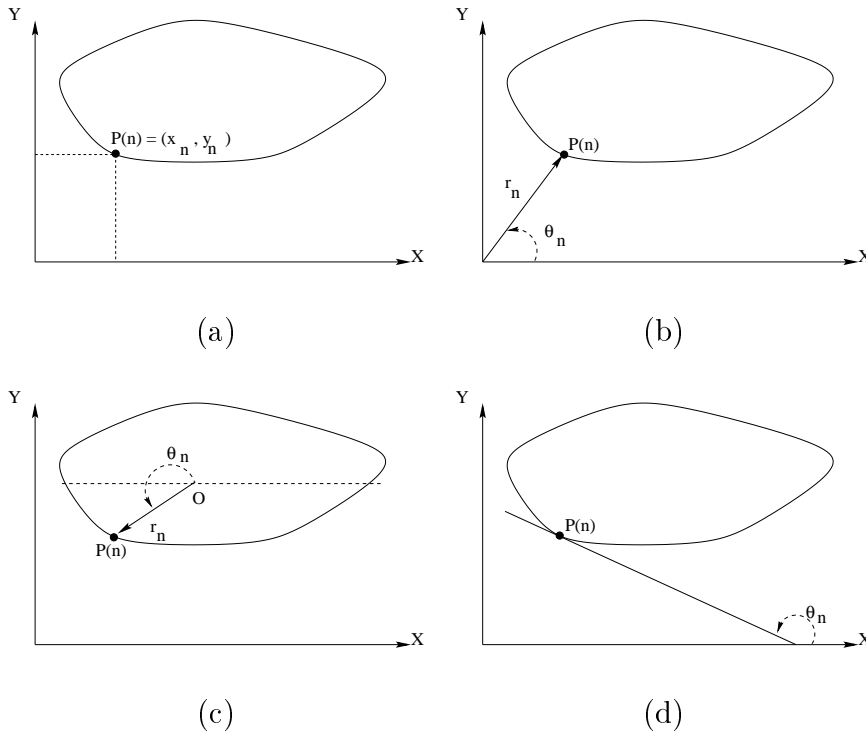


Figura 6.1: Sistemas de coordenadas: (a) cartesianas, (b) polares respecto del centro de coordenadas; (c) polares respecto del centroide y (d) tangencial.

La representación tangencial es una parametrización unidimensional del contorno (Loncaric 1998). Sin embargo, las demás representaciones basadas en los sistemas de coordenadas también permiten obtener parametrizaciones unidimensionales del contorno mediante el uso de funciones complejas, cuyo parámetro es la longitud de arco n (Granlund 1972; Persoon y Fu 1977; Richards y Hemami 1974). Por ejemplo, si se utilizan las coordenadas cartesianas, se tiene que

$$P(n) = x(n) + y(n)i = x_n + y_n i \quad (6.6)$$

6.2.2 Código de cadena

Un contorno puede ser representado por su “código de cadena” (Freeman 1961). Dados dos puntos consecutivos del contorno (x_i, y_i) , (x_{i+1}, y_{i+1}) , y dependiendo del tipo de conectividad empleado, sólo hay 4 u 8 posibles localizaciones para el punto (x_{i+1}, y_{i+1}) respecto del punto (x_i, y_i) (figuras 6.2(a) y (b)). Utilizando el código de cadena, un contorno puede ser representado por una secuencia de cambios de dirección si se recorre la curva en el sentido de las agujas del reloj a partir de un punto inicial (figuras 6.3(a) y (b)).

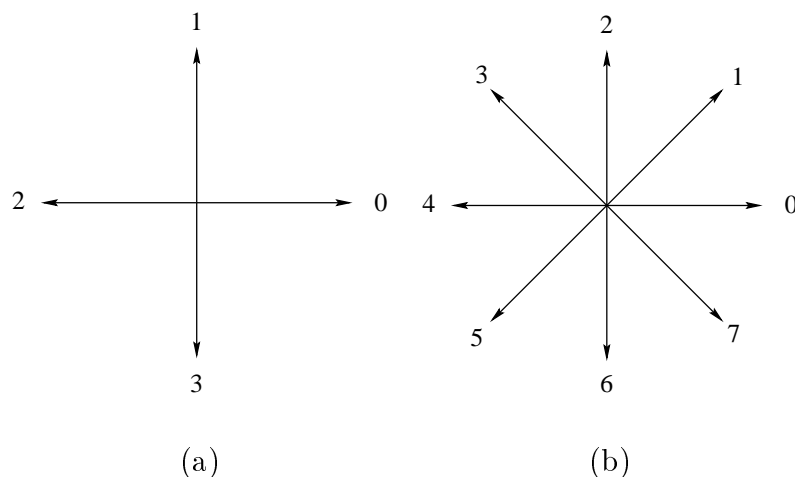


Figura 6.2: Direcciones de los códigos de cadena: (a) conectividad 4 y (b) conectividad 8.

El código de cadena permite obtener una representación compacta de un contorno y muchos algoritmos de análisis de formas pueden ser aplicados utilizando directamente el código de cadena (Freeman 1974; Davis 1986; Huang et al. 1999; Maravall 1993).

El código de la cadena depende del punto inicial, pero se puede normalizar si se elige como punto inicial a aquél que permita generar un código al cual le corresponda el número natural de menor valor. La localización del punto inicial es sencilla, ya que sólo hay que rotar de forma circular la secuencia de códigos

hasta encontrar el valor mínimo. Si el código de cadena de 8 direcciones es, por ejemplo, 3311006754, el código de cadena normalizado será 0067543311.

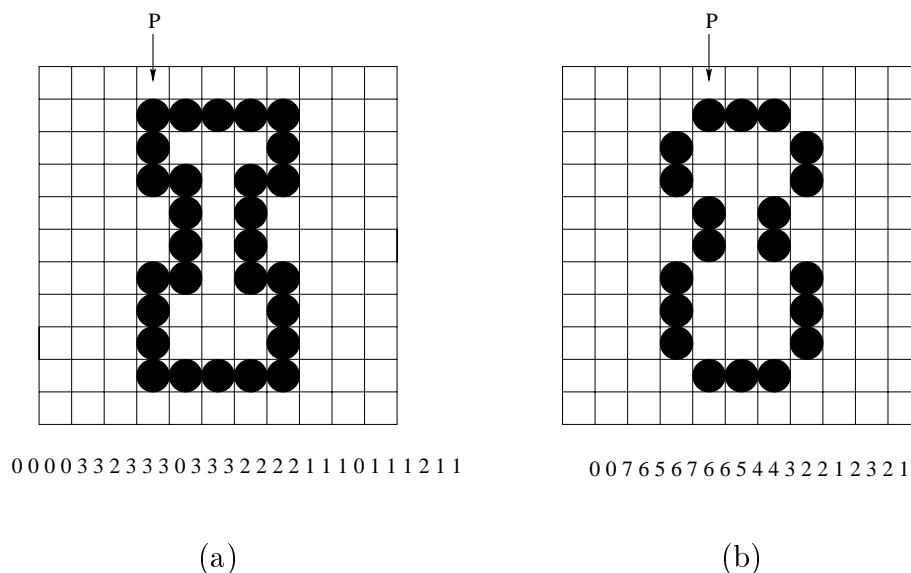


Figura 6.3: Códigos de cadena de (a) 4 direcciones y (b) 8 direcciones.

El código de cadena también depende de la rotación del contorno dentro de la imagen. Para superar este inconveniente, se puede utilizar, en cambio, “la primera diferencia del código de la cadena”, para lo cual sólo hay que contar, en el sentido contrario de las agujas del reloj, el número de direcciones que separan dos elementos adyacentes del código. Como se considera que el código es circular, el primer y último elementos son adyacentes entre sí. Por ejemplo, la primera diferencia del código de cadena de 8 direcciones 00765676654432212321 es 07771170770770711777.

También se ha propuesto una versión generalizada del código de cadena (Freeman y Saghri 1980; Saghri y Freeman 1981).

6.2.3 Signatura

La signatura¹ (*signature*) es una representación unidimensional que se puede definir como la curva que representa la distancia del centroide a cada punto del contorno como una función del ángulo.

Si el primer punto del contorno se elige utilizando el mismo criterio indicado para la representación basada en las coordenadas polares respecto del centroide,

¹Esta representación también es denominada “representación polar” (González-Jiménez 2000).

la signatura es también invariante a traslaciones y rotaciones. Así mismo también será invariante a cambios de escalas si se normaliza el tamaño del contorno.

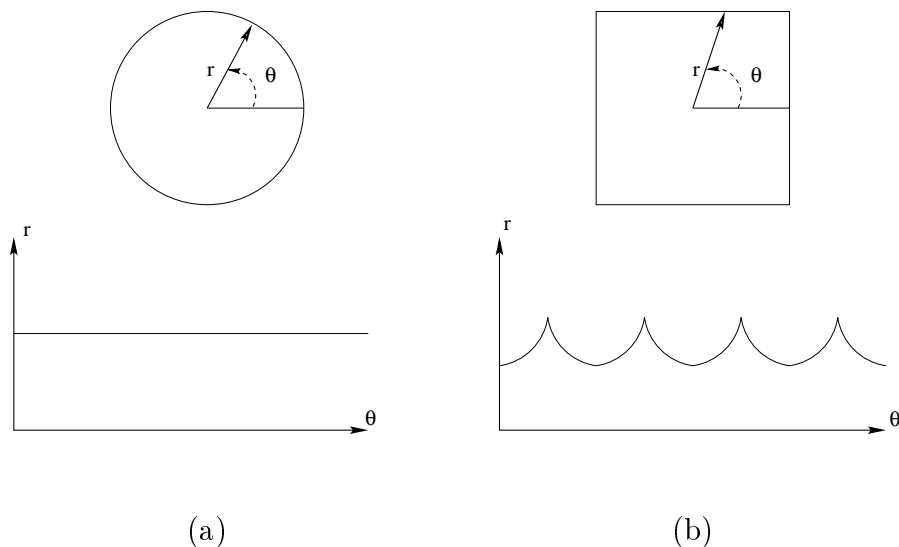


Figura 6.4: Contornos y signaturas de (a) un círculo y (b) un cuadrado.

Las posibles concavidades del contorno provocan que este método de calcular la signatura no sea muy adecuado, ya que puede provocar una representación multievaluada para algunos ángulos (figura 6.5(a)). Se puede evitar este problema si se considera la envolvente convexa del contorno y se descompone el contorno en subconjuntos convexos (figura 6.5(b)) (González-Jiménez 2000).

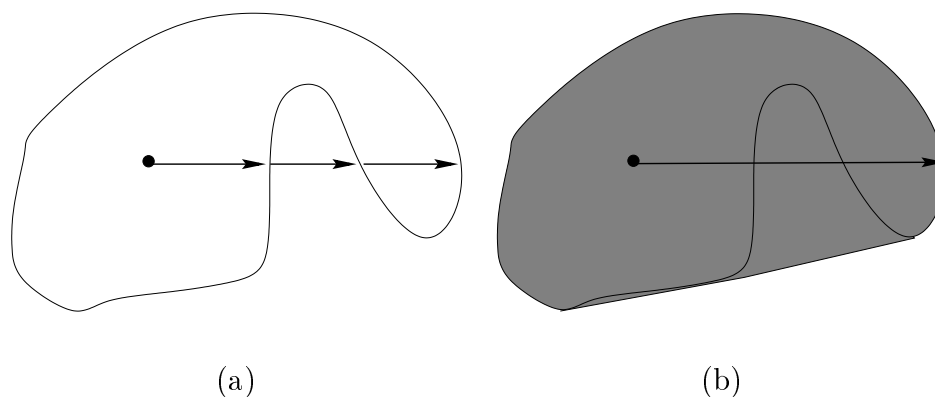


Figura 6.5: (a) Dificultades de la signatura: valores múltiples para un mismo ángulo. (b) Envolvente convexa del contorno (a) (González-Jiménez 2000).

Existen definiciones alternativas de la signatura que evitan este problema de multiplicidad de valores (figura 6.6):

- Distancia al punto opuesto: para cada punto A del contorno se calcula la distancia al punto B del contorno que esté situado en la recta normal al contorno en el punto A. La distancia obtenida es la signatura en el punto A (Sonka et al. 1993).

Las signaturas de A y B pueden ser diferentes, puesto que la recta normal al contorno en el punto B no tiene por qué pasar por A.

- Ángulo de la recta tangente: para cada punto A del contorno se calcula el ángulo de su recta tangente al contorno respecto de una recta de referencia (González y Woods 1996).

Esta definición de la signatura es equivalente a la representación tangencial descrita previamente (subsección 6.2.1).

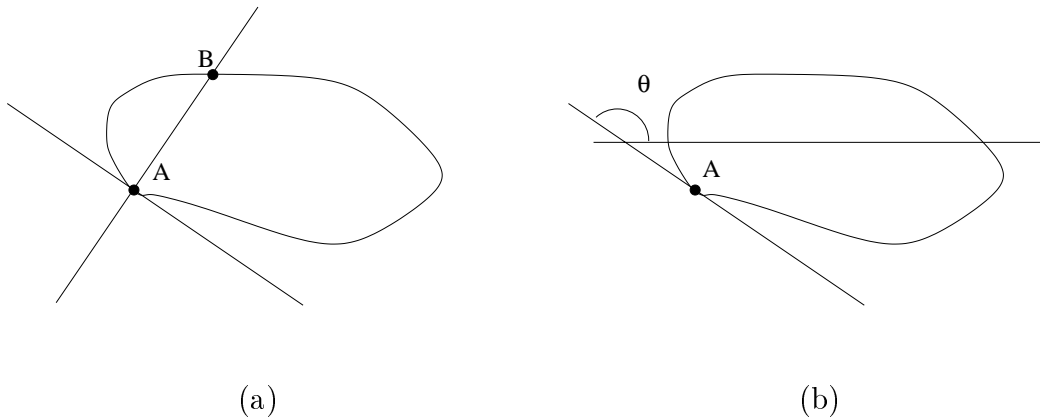


Figura 6.6: Definiciones alternativas de la signatura: (a) distancia al punto opuesto; (b) ángulo de la recta tangente respecto de una recta de referencia.

6.2.4 Método de la altura de arco

La función de altura de arco (*arc - height function*, *AHF*) permite generar una representación unidimensional del contorno (Lin et al. 1992; Loncaric 1998) utilizando una distancia calculada de la siguiente forma (figura 6.7):

- Se establece una longitud de arco de cuerda L .
- Para cada punto A del contorno se obtiene el punto B del contorno de forma que el arco de cuerda \overline{AB} tenga una longitud igual a L .
- Se determina el eje de simetría \overline{OC} que es perpendicular al arco de cuerda \overline{AB} .
- La longitud del segmento \overline{OC} es el valor de la función de altura de arco en el punto A.

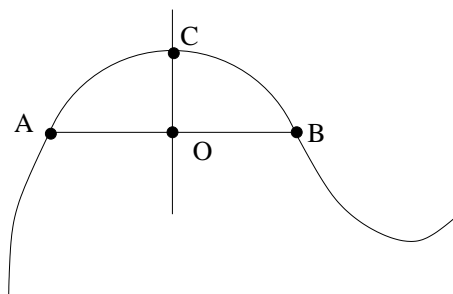


Figura 6.7: Concepto del altura de arco (Loncaric 1998).

6.2.5 Aproximaciones poligonales

Muchos algoritmos de reconocimiento no necesitan utilizar todos los puntos del contorno, sino sólo aquéllos que permitan obtener sus características esenciales. Las representaciones del contorno que sólo utilizan estos puntos característicos se denominan “aproximaciones del contorno” (*piecewise approximation*).

Una de las representaciones aproximadas más utilizadas es “la aproximación lineal o poligonal”, que genera un polígono cuyos vértices constituyen la representación del contorno original (Davis 1986; González y Woods 1996; González-Jiménez 2000; Loncaric 1998; Pitas 1993; Sonka et al. 1993).

La aproximación poligonal será exacta si se utilizan todos los puntos del contorno como vértices del polígono. Sin embargo, el objetivo es generar una aproximación poligonal con el menor número de vértices posible.

Se pueden generar diferentes polígonos a partir de un mismo contorno dependiendo del criterio que se utilice para medir el ajuste del polígono al contorno. Algunos de estos criterios son:

- Polígono de perímetro mínimo.
- Polígono interno de área máxima.
- Polígono externo de área mínima.

El uso de la aproximación poligonal para representar el contorno de un objeto ha sido propuesto en numerosas ocasiones (Bengtsson y Eklundh 1991; Pavlidis 1972; Pavlidis 1975; Pavlidis 1977; Ramer 1972; Wu y Leou 1993). Algunos de los métodos clásicos de obtención de la aproximación poligonal de un contorno son (González-Jiménez 2000; Loncaric 1998):

- Fusión: puntos consecutivos del contorno se van ajustando mediante una recta hasta que el error cometido supere un umbral preestablecido. A continuación se almacenan los parámetros de la recta y se repite el proceso con los

puntos del contorno restantes. Las intersecciones de las rectas constituyen los vértices del polígono.

- **División o partición:** se divide un segmento en dos hasta que se satisface un determinado criterio, como, por ejemplo, que la distancia de los puntos de un tramo del contorno al segmento que los aproxima no sobrepase un umbral determinado. Si la distancia de al menos un punto es superior al umbral, entonces se divide el segmento en dos eligiendo como nuevo vértice a aquél punto del contorno que diste más del segmento original. Si el contorno es cerrado, los puntos iniciales han de ser los que disten más entre sí.
- **División y fusión:** se combinan los dos métodos anteriores. Una estrategia posible consiste en dividir el contorno en segmentos hasta que el error cometido sea inferior a una cota prefijada. Al mismo tiempo, los segmentos poligonales pueden fusionarse si el segmento resultante permite obtener una aproximación al contorno cuyo error sea menor que el máximo permitido.

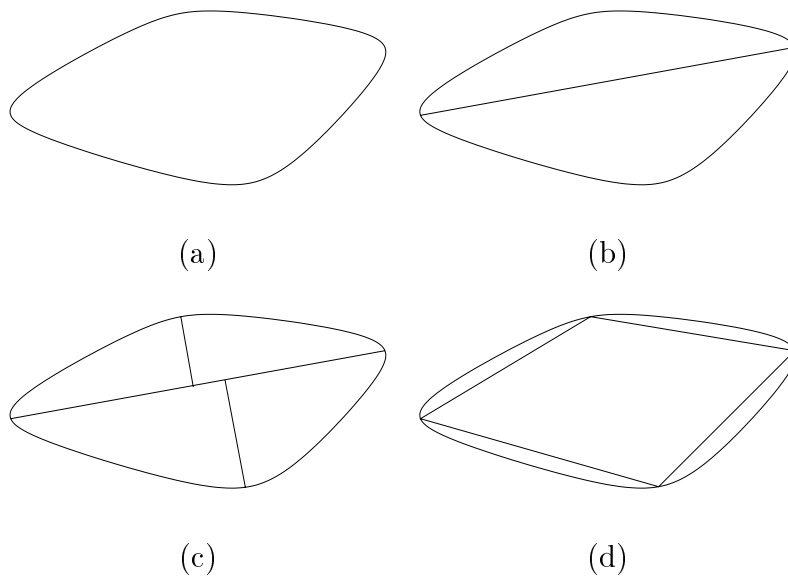


Figura 6.8: Fases de una aproximación poligonal de un contorno mediante división.

Una alternativa a estos métodos consiste en la utilización de puntos “característicos” del contorno como vértices del polígono. Estos puntos también son denominados “puntos críticos, dominantes o relevantes”.

Se han diseñado muchos métodos para encontrar los puntos críticos, los cuales se pueden clasificar en tres categorías (Fu y Yan 1997a; Fu y Yan 1997b):

- Métodos que buscan puntos críticos utilizando alguna medida distinta de la curvatura, como pueden ser la detección de ángulos, el estudio de simetrías

locales, etc. (Arcelli y Ramella 1993; Fischler y Bolles 1986; Ogawa 1989; Rosenberg 1972; Rosenberg 1974).

- Métodos que evalúan la curvatura mediante transformaciones del contorno en el espacio de escalas gaussiano (Asada y Brady 1986; Babaud et al. 1986; Cong y Ma 1998; Mokhtarian y Mackworth 1992; Mokhtarian 1995; Rattarangsi y Chin 1992; Ueda y Suzuki 1993; Witkin 1983). La principal ventaja de estos métodos radica en el hecho de que permiten extraer características de contornos ruidosos. Por el contrario, su complejidad computacional es mayor.
- Métodos que buscan los puntos críticos mediante la estimación de las curvaturas correspondientes (Chan et al. 2001; Davis 1977; Fairney y Fairney 1994; Rosenfeld y Johnston 1973; Rosenfeld y Weszka 1975; Teh y Chin 1989; Zhu y Chirlian 1995).

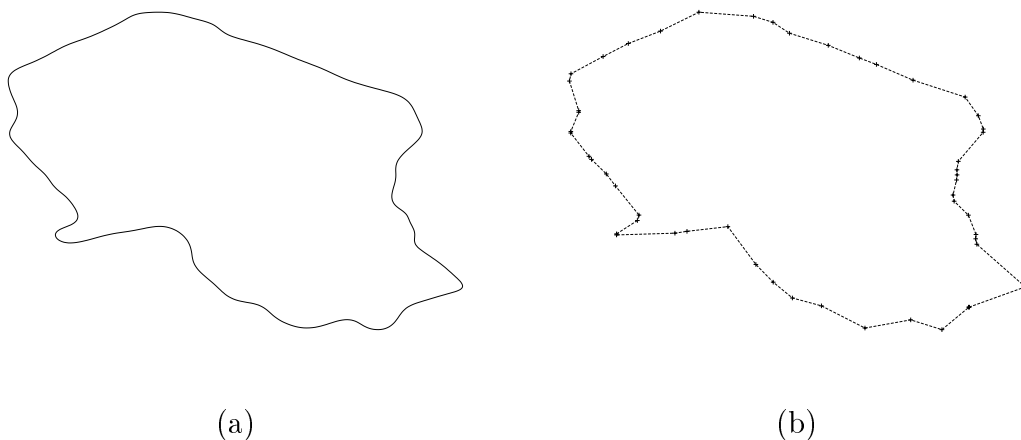


Figura 6.9: Aproximación poligonal usando puntos característicos: (a) contorno de la provincia de Córdoba suavizado con un filtro gaussiano ($\sigma = 5$); (b) aproximación poligonal de (a) en la que los vértices son los puntos de curvatura extrema obtenidos con el método de Mokhtarian y Mackworth (1.992).

Una estrategia diferente a las anteriores ha consistido en el uso de redes neuronales artificiales para construir una representación poligonal del contorno (Chung et al. 1994).

6.2.6 *Splines y B-splines*

Los polígonos permiten obtener una aproximación lineal de los contornos, ya que utilizan como elementos básicos a segmentos rectos. Se han diseñado otras representaciones aproximadas que utilizan otros tipos de elementos básicos, como

son los *splines* y los *B-splines* (Atkinson 1989; Bartels et al. 1991; de Boor 1978; Kincaid y Cheney 1991).

El uso de *splines* permite obtener una representación más ajustada al contorno (Ikebe y Miyamoto 1982) que las aproximaciones poligonales y, además, posee la propiedad de suavizar los cambios de curvatura. Su principal desventaja consiste en que las modificaciones locales del contorno afectan a la representación aproximada mediante *splines*. Este problema de sensibilidad local ha sido resuelto mediante el uso de *B-splines*, ya que evitan que las modificaciones locales del contorno afecten a las demás partes de la representación construida (Cohen et al. 1995; Gu y Tjahjadi 2000).

6.2.7 Representaciones jerárquicas

Algunas veces es necesario descomponer el contorno en dos o más partes para facilitar su reconocimiento, por ejemplo, cuando presenta concavidades o zonas ocultas. Las representaciones jerárquicas permiten establecer una relación estructural entre las partes en que se divide un contorno.

Algunas de las representaciones jerárquicas están basadas en los siguientes conceptos:

- Envolverte convexa (*convex hull*).
- Árbol de tira (*strip tree*).
- Gramáticas formales.

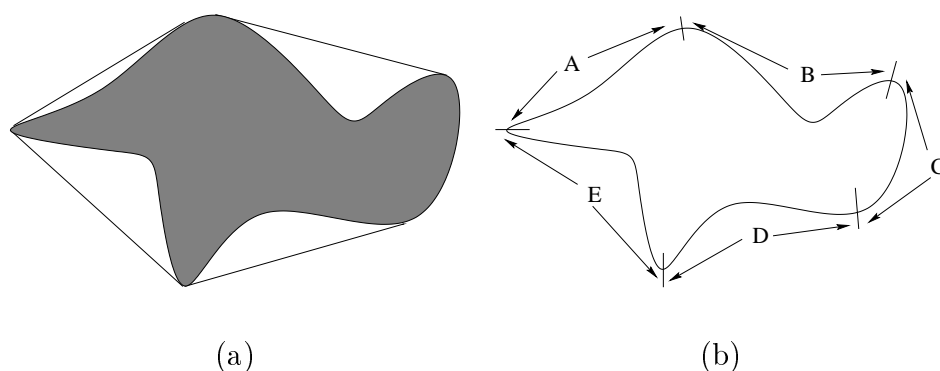


Figura 6.10: (a) Envolverte convexa de un contorno. (b) Descomposición del contorno.

La envolverte convexa de un contorno se define como el objeto convexo más pequeño que contiene al contorno. Un objeto es convexo si el segmento que

que dos puntos cualesquiera de dicho objeto está contenido completamente en el objeto.

Existen diferentes algoritmos que permiten calcular la envolvente convexa de un contorno (Davies 1990). La descomposición del contorno en sus componentes convexas permite obtener una representación jerárquica que facilita la descripción del objeto (González-Jiménez 2000).

La representación mediante “árbol de tiras” (Ballard 1981) descompone el contorno en diferentes partes utilizando rectángulos que se ajustan al contorno (figura 6.11).

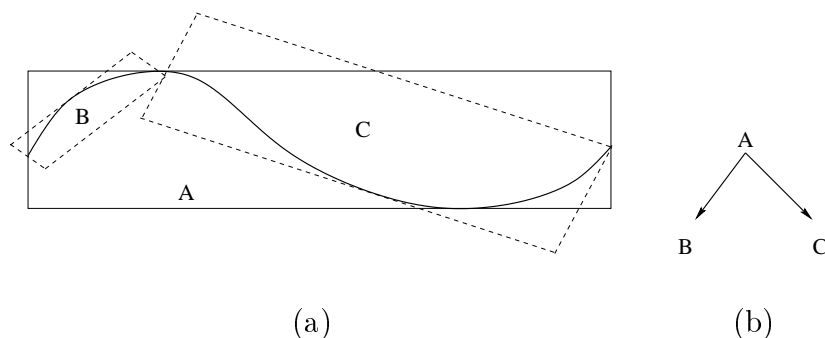


Figura 6.11: Representación mediante tiras: (a) primera descomposición de un contorno y (b) árbol de tiras de (a) (Davis 1986).

La teoría de las gramáticas formales fue establecida por Noam Chomsky en 1956 (Aho et al. 1990; Chomsky 1956) y ha sido utilizada en múltiples aplicaciones incluyendo la representación jerárquica de contornos (Fu y Mui 1981; Henderson 1983; Pavlidis y Chen 1979).

Las gramáticas formales permiten introducir diferentes niveles de abstracción en la representación jerárquica de los contornos, porque los elementos de los niveles inferiores se corresponden con partes pequeñas o geoméricamente básicas del contorno, mientras que los elementos de los niveles superiores se corresponden con partes más grandes o geoméricamente más complejas (Davis 1986).

6.3 Descriptores de un contorno

Una vez obtenida la representación de un contorno, se ha de proceder a extraer u obtener un conjunto de descriptores o atributos, denominado “vector de características”, que permita su análisis.

Las propiedades que deben poseer los descriptores, atributos o características son (Castleman 1979):

- Discriminación: las características deberían tomar valores significativamente diferentes para objetos de clases diferentes.
- Fiabilidad: las características deberían tomar valores similares para objetos de la misma clase.
- Independencia: las características no deberían estar correlacionadas.
- Número pequeño: la complejidad de un sistema de reconocimiento se incrementa a medida que aumenta el número de características que deben ser usadas.

Además, los descriptores deben ser invariantes a traslaciones, rotaciones y cambios de escala del contorno.

Los descriptores o atributos del contorno pueden ser clasificados como (Pratt 1978; Pratt 1991):

- Atributos métricos.
- Atributos topológicos.
- Atributos analíticos.

6.3.1 Atributos métricos

Los atributos métricos están basados en una medida de distancia d entre puntos que posee las siguientes características:

- $d : R^2 \times R^2 \longrightarrow R_+$

$$d((x_1, y_1), (x_2, y_2)) \geq 0 \quad (6.7)$$

- Axioma de separación:

$$d((x_1, y_1), (x_2, y_2)) = 0 \Leftrightarrow (x_1, y_1) = (x_2, y_2) \quad (6.8)$$

- Axioma de simetría

$$d((x_1, y_1), (x_2, y_2)) = d((x_2, y_2), (x_1, y_1)) \quad (6.9)$$

- Desigualdad triangular

$$d((x_1, y_1), (x_3, y_3)) \leq d((x_1, y_1), (x_2, y_2)) + d((x_2, y_2), (x_3, y_3)) \quad (6.10)$$

Las medidas de distancias más utilizadas en el análisis de imágenes son

- Distancia euclídea:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6.11)$$

- Distancia de la suma de los valores absolutos, distancia d_4 , distancia de la ciudad de los bloques (*city block distance*) o distancia de Manhattan

$$d((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2| \quad (6.12)$$

- Distancia del máximo de los valores absolutos, distancia d_8 o distancia de ajedrez (*chess-board distance*)

$$d((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\} \quad (6.13)$$

Una vez establecida la distancia métrica que va a ser utilizada, pueden obtenerse los atributos métricos del contorno de un objeto, como, por ejemplo:

- Perímetro

$$P = \sum_{i=1}^N d((x_i, y_i), (x_{i+1}, y_{i+1})) \quad (6.14)$$

donde N es el número de puntos de la forma y $(x_{N+1}, y_{N+1}) = (x_1, y_1)$

- Área

$$A = N \quad (6.15)$$

donde N es el número de puntos del objeto, es decir, delimitados por el contorno.

- Circularidad² o compacticidad

$$C = \frac{P^2}{A} \quad (6.17)$$

Este atributo es adimensional y tiene un valor mínimo de 4π para los círculos.

- Rectangularidad

$$R = \frac{A}{A_R} \quad (6.18)$$

donde A es el área del objeto delimitado por el contorno y A_R es el área del mínimo rectángulo abarcador del contorno. Este atributo adimensional tiene un valor máximo de 1 para objetos rectangulares.

²Existe un atributo similar a la circularidad denominado “proporción de delgadez” (*thinness ratio*) cuya definición es (Duda y Hart 1973):

$$T = 4\pi \frac{A}{P^2} \quad (6.16)$$

Una medida alternativa es (Castleman 1979)

$$R' = \frac{W}{L} \quad (6.19)$$

donde W y L son la anchura y la altura, respectivamente, del mínimo rectángulo abarcador del contorno. Este atributo permite distinguir objetos delgados o alargados de objetos similares a cuadrados o círculos.

Los atributos métricos son invariantes a rotaciones y traslaciones.

6.3.2 Atributos topológicos

Los atributos topológicos describen propiedades que son invariantes a deformaciones continuas de los objetos, como, por ejemplo, los estrechamientos o alargamientos.

Los atributos topológicos suelen ser utilizados como descriptores de regiones y no de contornos.

La conectividad es un atributo topológico, puesto que no existe ninguna deformación continua que provoque un aumento o una disminución de la conectividad. Así mismo, el número de huecos de un objeto también es una propiedad topológica.

El número de Euler permite definir una relación entre las componentes conexas y los huecos de un objeto (figura 6.12):

$$E = C - H \quad (6.20)$$

donde C es el número de componentes conexas y H el número de huecos de un objeto.

6.3.3 Atributos analíticos

Los atributos analíticos de un contorno son descriptores que se obtienen mediante cálculos matemáticos que no utilizan una medida de distancia.

Algunos de los atributos analíticos son:

- Curvatura.
- Energía (*bending energy*).
- Momentos.
- Descriptores de Fourier.

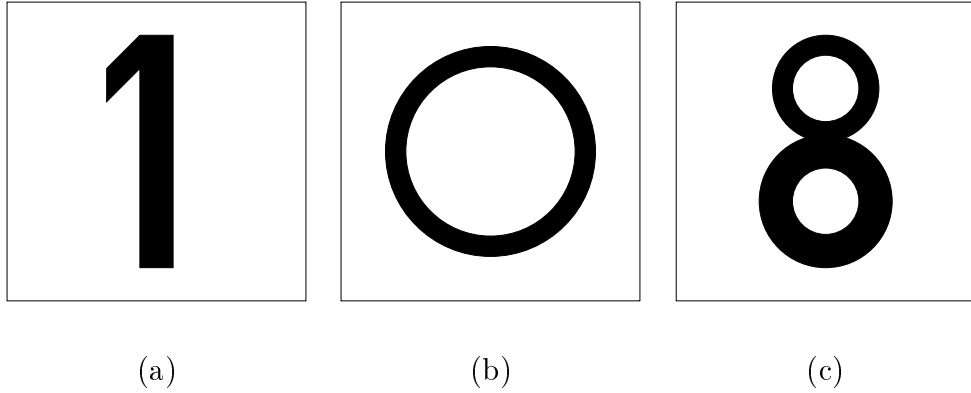


Figura 6.12: Números de Euler de varios objetos: (a) $E = 1$; (b) $E = 0$; (c) $E = -1$.

Curvatura

Si los puntos de una curva son parametrizados en función de la longitud de arco n de la forma:

$$r(n) = (x(n), y(n)) \quad (6.21)$$

entonces el ángulo tangente a la curva en un punto se define como

$$\phi(n) = \arctan\left(\frac{y'(n)}{x'(n)}\right) \quad (6.22)$$

y la curvatura como

$$\kappa(n) = \phi'(n) \quad (6.23)$$

verificándose las siguientes equivalencias (Pogorelov 1987)

$$\kappa(n)^2 = |r''(n)|^2 = \frac{(x''(n)y'(n) - y''(n)x'(n))^2}{(x'(n)^2 + y'(n)^2)^3} \quad (6.24)$$

Los puntos de curvatura cero y extrema (máxima o mínima) han sido utilizados como descriptores de los contornos por diferentes algoritmos de reconocimiento de objetos (Chan et al. 2001; Davis 1977; Fairney y Fairney 1994; Gu y Tjahjadi 2000; Rosenfeld y Johnston 1973; Rosenfeld y Weszka 1975; Teh y Chin 1989; Zhu y Chirlian 1995).

Además, las transformaciones del contorno mediante filtros gaussianos han permitido diseñar algoritmos de reconocimiento multiescala basados en la curvatura (Asada y Brady 1986; Babaud et al. 1986; Cong y Ma 1998; Mokhtarian y Mackworth 1992; Mokhtarian 1995; Rattarangsi y Chin 1992; Ueda y Suzuki 1993; Witkin 1983).

Energía (*bending energy*)

La energía se define como (Canhan 1970; Young et al. 1974)

$$E(n) = \frac{1}{L} \int_0^n \kappa(n)^2 dn \quad (6.25)$$

donde

- κ es la función de curvatura,
- n el parámetro de longitud de arco
- y L es la longitud total de la curva.

La versión discreta de la energía se define como

$$E = \frac{1}{L} \sum_{i=0}^{N-1} \kappa(i)^2 \quad (6.26)$$

donde N es el número de puntos de la curva.

En la actualidad, la energía no se calcula directamente, sino que se obtiene utilizando los descriptores de Fourier y el teorema de Parseval (Jain 1989; Loncaric 1998; Oppenheim et al. 1983; Papoulis 1991; Pitas 1993; Sonka et al. 1993):

$$E = \sum_{k=0}^{N-1} |Z(k)|^2 \left(\frac{2\pi k}{L} \right)^4 \quad (6.27)$$

donde $Z(k)$ ($k = 0, 1, \dots, N-1$) son los descriptores de Fourier (ecuación 6.48).

El círculo posee la energía mínima entre todas las curvas con igual perímetro L :

$$E_{\text{círculo}} = \frac{4\pi^2}{L^2} \quad (6.28)$$

Los inconvenientes de la energía son:

- No permite la reconstrucción del contorno.
- Su valor depende del perímetro.

Se puede normalizar el valor de la energía en el rango $[0, 1]$ mediante la expresión (Levine 1985; Pitas 1993)

$$E_N = 1 - \frac{E_{\text{círculo}}}{E_{\text{objeto}}} = 1 - \frac{4\pi^2}{L \sum_{i=0}^{N-1} |k(i)|^2} \quad (6.29)$$

La energía ha sido utilizada junto con el código de cadena para el reconocimiento de células (Smeulders et al. 1980).

Momentos

Se define el momento general de orden p, q como (Hu 1962)

$$m_{pq} = \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (6.30)$$

La versión discreta para imágenes digitales es

$$m_{pq} = \sum_{x=0}^{N_1} \sum_{y=0}^{N_2} x^p y^q f(x, y) \quad (6.31)$$

donde $p, q \in \{0, 1, \dots, \infty\}$.

Para conseguir la invarianza respecto de las traslaciones, se calculan los momentos centrales que se definen como

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (6.32)$$

donde (\bar{x}, \bar{y}) son las coordenadas del centro de gravedad o centroide del objeto

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad (6.33)$$

$$\bar{y} = \frac{m_{01}}{m_{00}} \quad (6.34)$$

Los momentos invariantes a rotaciones se definen en función de los momentos centrales (Maravall 1993):

$$\phi_{pq} = \sum_{r=0}^p \sum_{s=0}^q (-1)^{q-s} r^p s^q (\cos(\theta))^{p-r+s} (\sin(\theta))^{q-s+r} \mu_{p-r+q-s, r+s} \quad (6.35)$$

donde θ es el ángulo del eje de mínima inercia del objeto 6.44.

Para definir los momentos invariantes a la escala es necesario definir los momentos centrales normalizados (Escalera 2001)

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (6.36)$$

donde

$$\gamma = \frac{p+q}{2} + 1 \quad \wedge \quad p+q \geq 2 \quad (6.37)$$

Si una imagen $f(x, y)$ es binaria, el cálculo de los momentos de orden p, q se simplifica

$$m_{pq} = \sum_x \sum_y x^p y^q \quad (6.38)$$

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \quad (6.39)$$

Los momentos generales que se definen a partir de un contorno permiten caracterizarlos adecuadamente, pero son muy sensibles al ruido y a pequeñas variaciones en la forma del contorno, siendo necesario utilizar un mayor número de momentos para obtener una descripción más robusta. Por el contrario, los momentos generales basados en la región delimitada por el contorno son más robustos (Maravall 1993).

Algunos descriptores de los objetos binarios se pueden definir a partir de sus momentos (Pitas 1993):

- El área es igual al momento m_{00} .
- El centro de gravedad o centroide

$$C = (\bar{x}, \bar{y}) \quad (6.40)$$

donde los valores de \bar{x} e \bar{y} se obtienen a partir de las ecuaciones 6.33 y 6.34. Si el área de un objeto R es de N puntos, entonces las coordenadas del centro de gravedad pueden obtenerse de una forma alternativa:

$$\bar{x} = \frac{1}{N} \sum_{(x,y) \in R} x \quad (6.41)$$

$$\bar{y} = \frac{1}{N} \sum_{(x,y) \in R} y \quad (6.42)$$

- El ángulo del eje de mínima inercia o ángulo de orientación del objeto se obtiene minimizando la función

$$S(\theta) = \sum_{(x,y) \in R} ((x - \bar{x})\cos(\theta) - (y - \bar{y})\sin(\theta))^2 \quad (6.43)$$

que alcanza el valor mínimo para

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (6.44)$$

- Excentricidad del objeto

$$\epsilon = \left(\frac{\mu_{02}\cos(\theta)^2 + \mu_{20}\sin(\theta)^2 - \mu_{11}\sin(2\theta)}{\mu_{02}\sin(\theta)^2 + \mu_{20}\cos(\theta)^2 + \mu_{11}\cos(2\theta)} \right)^2 \quad (6.45)$$

El uso de los momentos fue propuesto por Hu (1.962), que comprobó que la descripción de la forma basada en los momentos permite preservar la información. Los momentos m_{pq} (ecuación 6.30) se definen de forma unívoca a partir de la función $f(x, y)$ y viceversa, los momentos m_{pq} son suficientes para reconstruir la función original $f(x, y)$.

Los momentos pueden ser generalizados sustituyendo los operadores de transformación $x^p y^q$ por polinomios $P_p(x) P_q(y)$. Si los polinomios son ortogonales (Teague 1980), la información redundante producida por los momentos es mínima (Prokop y Reeves 1993).

Algunas de las ventajas del uso de los momentos para el análisis de formas son:

- Poseen una definición matemática precisa.
- Los primeros momentos se corresponden con características de las formas como el área (momento m_{00}), el centro de gravedad o centroide (ecuaciones 6.33 y 6.34), el ángulo del eje de mínima inercia (ecuación 6.44) o la excentricidad (ecuación 6.45).

Sin embargo, también presenta algunas desventajas (Loncaric 1998):

- Es difícil establecer una correspondencia entre los momentos de órdenes superiores y las características de las formas.
- No permiten detectar características locales de las formas.

Descriptores de Fourier

Si un contorno cerrado es parametrizado en función de la longitud de arco n como

$$r(n) = (x(n), y(n)) \quad (6.46)$$

entonces se puede definir la siguiente función compleja

$$z(n) = x(n) + y(n) i \quad n = 0, 1, \dots, N - 1 \quad (6.47)$$

y su transformada discreta de Fourier

$$Z(k) = \frac{1}{N} \sum_{n=0}^{N-1} z(n) e^{\frac{-2\pi i n k}{N}} \quad (6.48)$$

Los coeficientes $Z(k)$ se denominan “descriptores de Fourier” y poseen algunas propiedades interesantes (Pitas 1993; Sonka et al. 1993):

- Permiten reconstruir la curva original

$$z(n) = \frac{1}{N} \sum_{k=0}^{N-1} Z(k) e^{\frac{2\pi i n k}{N}} \quad (6.49)$$

- $Z(0)$ es el centro de gravedad de la curva.
- Las magnitudes de los coeficientes $|Z(k)|$ para $k = 0, 1, \dots, N - 1$ son invariantes a rotaciones.
- Las magnitudes de los coeficientes $|Z(k)|$ para $k = 1, 2, \dots, N - 1$ son invariantes a traslaciones.
- Los valores de los ángulos $\arg(Z(k))$ para $k = 0, 1, \dots, N - 1$ son invariantes a cambios de escala.
- El error cuadrático medio de las magnitudes puede ser usado como medida de discrepancia entre dos curvas

$$E = \sum_{k=0}^{N-1} (|Z_1(k)| - |Z_2(k)|)^2 \quad (6.50)$$

Además, los descriptores de Fourier pueden ser utilizados para calcular el centroide y los momentos de segundo orden (Kiryati y Maydan 1989; Sonka et al. 1993).

6.4 Algoritmos de reconocimiento de contornos

Se han diseñado muchos algoritmos de reconocimiento de formas basados exclusivamente en el contorno de los objetos (Loncaric 1998; van Otterloo 1991; Pavlidis 1978; Pavlidis 1980).

A continuación se van a citar algunos de los algoritmos de reconocimiento de contornos diseñados. Obviamente, no se pretende en modo alguno efectuar una revisión exhaustiva, sino mostrar cómo el problema de reconocimiento de contornos puede ser observado desde muchos puntos de vista diferentes pero no excluyentes.

Algunos de los algoritmos de reconocimiento de contornos, clasificados según los fundamentos teóricos, son:

- Puntos característicos, críticos, dominantes o relevantes (Cong y Ma 1998; Fairney y Fairney 1994; Fu y Yan 1997a; Fu y Yan 1997b; Freeman 1978; Gu y Tjahjadi 2000; Huang et al. 1999; Liu y Srinath 1990; Mokhtarian y Mackworth 1992; Mokhtarian 1995).

- Descriptores de Fourier (Cheng et al. 1996; Granlund 1972; Lin y Chellapa 1987; Persoon y Fu 1977; Richards y Hemami 1974; Wallace y Wintz 1980; Zahn y Roskies 1972)
- Momentos (Belkasim et al. 1991; Flusser y Suk 1993; Prokop y Reeves 1993; Weiss 1993).
- Transformada de Hough (Ballard y Brown 1981; Duda y Hart 1972; Guil et al. 1999; Hough 1962).
- Redes neuronales artificiales (Bebis et al. 1999; Fu y Yan 1997c; Eom 1998; Rishikesh y Venkatesh 2001).
- Wavelet (Chuang y Kuo 1996; Wunsch y Laine 1995).
- Etc.

Parte III

Evaluación y Diseño

Capítulo 7

Análisis de las medidas de evaluación

7.1 Introducción

El sistema de reconocimiento de objetos que se ha diseñado utiliza un algoritmo de detección de bordes en imágenes en color. Para elegir el detector de bordes más adecuado, es necesario comparar diferentes algoritmos mediante el uso de alguna medida de evaluación o de calidad (capítulo 8). Sin embargo, antes de proceder a tal estudio comparativo, se ha considerado necesario seleccionar la medida de evaluación basada en la discrepancia que tenga un mejor rendimiento.

Para seleccionar la medida de calidad, se han aplicado diferentes detectores de bordes en color a imágenes artificiales y se han analizado los resultados que cada medida ha considerado como óptimos para cada detector.

Las medidas de calidad que han sido analizadas son:

- Porcentaje de discrepancia (Bryant y Bouldin 1979; Peli y Malah 1982; Peli 1979; Ramesh y Haralick 1992a; Ramesh y Haralick 1992b; Román-Roldán et al. 2001; Zhang 1996; Zhu et al. 1999).
- Medida de Baddeley (1992a,1992b).
- Medida de Pratt (Abdou y Pratt 1979; Pratt 1978; Pratt 1991).
- Medida de Venkatesh y Kitchen (1992).

La medida del porcentaje de discrepancia y la medida de Baddeley son descritas en las secciones 7.2 y 7.3, respectivamente. Las medidas de Pratt y de Venkatesh y Kitchen ya fueron descritas en las subsecciones 4.3.2 y 4.3.3.

En la sección 7.4 se indican las modificaciones efectuadas en las medidas de calidad para que puedan ser aplicadas a cualquier tipo de borde. En particular, se ha de indicar que se ha diseñado una variante general del método de evaluación de Venkatesh y Kitchen que puede ser aplicada a imágenes con bordes de cualquier forma geométrica.

Por último, la sección 7.5 muestra el proceso realizado para seleccionar la medida de calidad más adecuada para el estudio comparativo de los detectores de bordes en color.

7.2 Porcentaje de discrepancia

Al comparar dos imágenes binarias, se pueden producir los siguientes errores:

- Falso positivo: se considera erróneamente que un punto pertenece al borde.
- Falso negativo: se considera erróneamente que un punto no pertenece al borde.

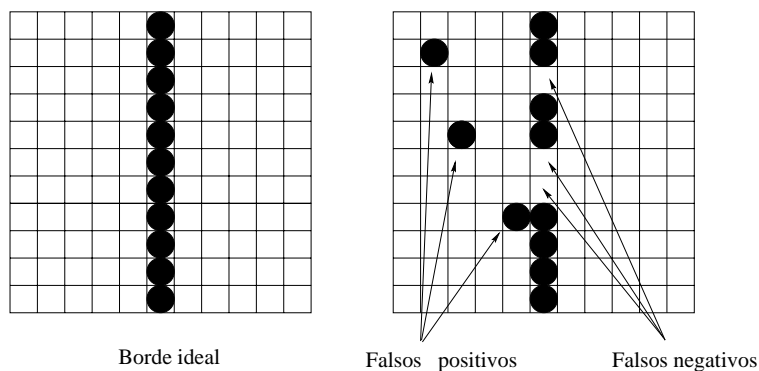


Figura 7.1: Puntos de discrepancia.

Se define el porcentaje de discrepancia como (Bryant y Bouldin 1979; Peli y Malah 1982; Peli 1979; Ramesh y Haralick 1992a; Ramesh y Haralick 1992b; Román-Roldán et al. 2001; Zhang 1996; Zhu et al. 1999)

$$D = \frac{fp + fn}{N} \times 100 \quad (7.1)$$

donde

- fp es el número de puntos que han sido considerados como falsos positivos,
- fn es el número de puntos que han sido considerados como falsos negativos

- y N es el número total de puntos de la imagen.

El porcentaje de discrepancia varía entre 0 (valor óptimo) y 100 (valor pésimo).

El principal inconveniente de esta medida es que no tiene en cuenta la distancia de los falsos positivos al borde verdadero.

7.3 Medida de Baddeley

Baddeley (1992a,1992b) propuso una medida de error para imágenes binarias basada en “la distancia de Hausdorff”. Dados dos subconjuntos $A, B \subseteq X$, se define la distancia de Hausdorff entre A y B como:

$$H(A, B) = \max\left\{\sup_{x \in A} d(x, B), \sup_{x \in B} d(x, A)\right\} \quad (7.2)$$

donde $d(x, A)$ denota la distancia más corta de $x \in X$ a $A \subseteq X$:

$$d(x, A) = \inf\{\rho(x, a) : a \in A\} \quad (7.3)$$

siendo ρ una distancia métrica entre puntos.

El valor de $H(A, B)$ representa la máxima distancia entre un punto de un conjunto y el punto más cercano del otro conjunto.

La medida propuesta por Baddeley fue:

$$\Delta_w^p(A, B) = \left[\frac{1}{N} \sum_{x \in X} |w(d(x, A)) - w(d(x, B))|^p \right]^{1/p} \quad (7.4)$$

donde

- $d(x, A)$ denota la distancia más corta de $x \in X$ a $A \subseteq X$ (ecuación 7.3)
- w es una función continua, cóncava y estrictamente creciente en 0.
- y $1 \leq p < \infty$.

Si A representa el borde de la imagen ideal y B el de la imagen generada por un detector de bordes, $\Delta_w^p(A, B)$ no sólo tiene en cuenta los puntos de los bordes de la imagen ideal y de la imagen a evaluar, sino que también considera los puntos que no pertenecen a ninguno de los dos bordes.

Baddeley utilizó como función w la siguiente transformación

$$w(t) = \min\{t, c\} \quad (7.5)$$

donde $c > 0$ es un valor fijo preestablecido.

El uso de la constante c permite:

- limitar el valor de $\Delta_w^p(A.B)$ entre 0 y c ,
- establecer una cota de error en la distancia de cada punto x a la imagen A o B
- y, además, descartar la influencia de los puntos x que estén situados más allá de c unidades de A y B , ya que no contribuyen al valor final de la medida de error.

El valor de p determina la importancia relativa de los errores de localización.

Baddeley utilizó los valores $c = 5$ y $p = 2$ para comparar su nueva medida con la medida de Pratt (Baddeley 1992a; Baddeley 1992b).

7.4 Variantes o extensiones de las medidas de evaluación

La medida de Pratt y la medida de Venkatesh y Kitchen han sido generalizadas para poder ser utilizadas con cualquier tipo de borde, independientemente de su forma geométrica.

La medida de error del porcentaje de discrepancia se puede extender de forma inmediata a cualquier tipo de borde, como se puede observar en la figura 7.2.

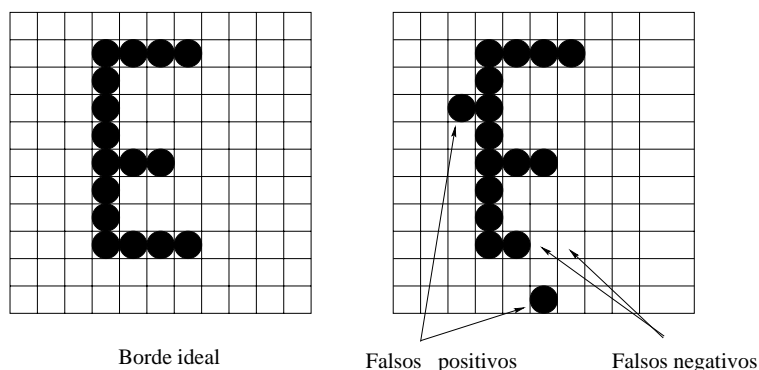


Figura 7.2: Puntos de discrepancia de dos imágenes arbitrarias.

La medida de Baddeley tampoco necesita ninguna extensión, ya que puede ser utilizada para comparar dos imágenes binarias cualesquiera. Los parámetros que se han utilizado en las pruebas son $c = 5$ y $p = 2$.

7.4.1 Variante de la medida de Pratt

La medida de Pratt (Abdou y Pratt 1979; Pratt 1978) fue diseñada para bordes verticales o inclinados (subsección 4.3.2), pero también puede ser aplicada a cualquier imagen si se considera la siguiente definición alternativa:

$$F = \frac{1}{\text{máximo}\{n(I), n(D)\}} \sum_{x \in D} \frac{1}{1 + \alpha \tilde{d}(x)^2} \quad (7.6)$$

donde

- $n(I)$ es el número de puntos del borde ideal,
- $n(D)$ es el número de puntos de borde detectados,
- α es una constante de escala, que pondera los errores producidos por bordes gruesos o desplazados. En las pruebas realizadas se han utilizado los valores $\alpha = \frac{1}{9}$ y $\alpha = 1.0$
- y $\tilde{d}(x)$ es la distancia entre un punto de borde y el borde ideal definida como

$$\tilde{d}(x) = \min\{d(x, I), c\} \quad (7.7)$$

siendo $d(x, I)$ la distancia más pequeña de x al borde ideal I (ecuación 7.3) y $c > 0$.

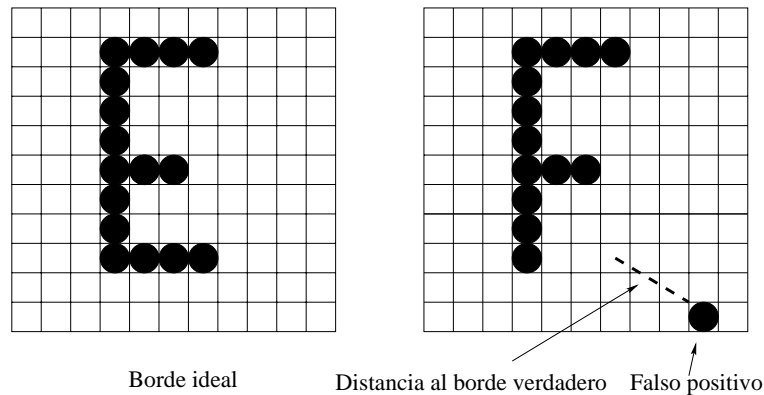


Figura 7.3: Variante de la medida de Pratt.

La diferencia con la versión original propuesta por Pratt se encuentra en el uso de la función $\tilde{d}(x)$, que limita la distancia máxima de un punto al borde verdadero a c unidades. La constante $c > 0$ establece una cota superior en la distancia de error. Se ha utilizado el valor $c = 5$.

7.4.2 Versión general del método de Venkatesh y Kitchen

Se ha diseñado una versión general del método de Venkatesh y Kitchen (1.992) (véase la subsección 4.3.3), que está basada en los siguientes conceptos:

- La “tira de detección” del método original se sustituye por una “zona de detección” que consta de dos partes (figura 7.4):
 - Zona ideal: está compuesta por los puntos del borde ideal.
 - Zona de detección múltiple: engloba a los puntos que son vecinos de algún punto del borde ideal.

El resto de la imagen se denomina “zona de falsos positivos”.

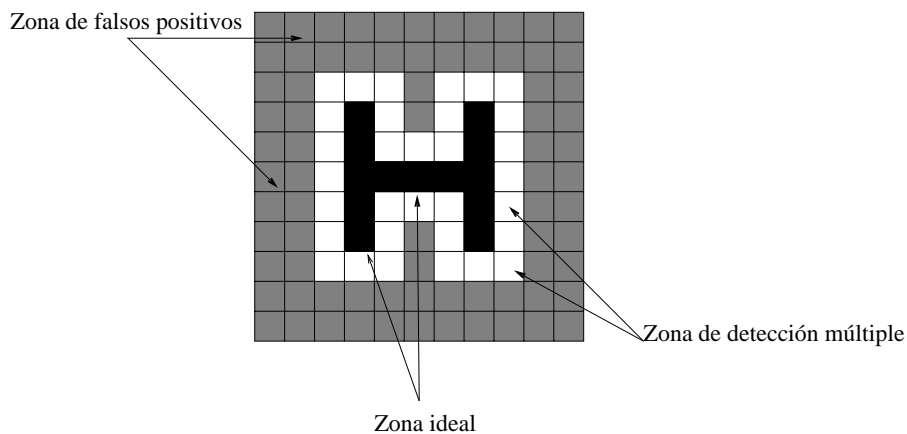


Figura 7.4: Zona de detección correspondiente a un borde cualquiera.

- Para poder calcular las medidas de error, se establecen las siguientes definiciones:
 - Falso negativo: un punto se considera como falso negativo si es un punto de borde de la imagen ideal que no ha sido detectado en la imagen a evaluar y, además, el número de vecinos que tiene en la zona de detección de la imagen a evaluar es menor que los que tiene en la imagen ideal.

La medida de error de los falsos negativos se calcula como:

$$FN = \frac{fn}{n(ideal)} \quad (7.8)$$

donde fn es el número total de falso negativos y $n(ideal)$ es el número de puntos del borde ideal.

- Falso positivo: un punto se considera como falso positivo si es detectado fuera de la zona de detección.

La medida de error de los falsos positivos se calcula como:

$$FP = \frac{fp}{n(zona_{fp})} \quad (7.9)$$

donde fp es el número total de falsos positivos y $n(zona_{fp})$ es el número total de puntos de la zona de falsos positivos.

- Detección múltiple: un punto se considera como punto de detección múltiple si es detectado en la zona de detección múltiple y, además, el número de vecinos que tiene en la zona de detección¹ de la imagen a evaluar es mayor que los que tiene en la imagen ideal.

La medida de error de la detección múltiple se calcula como:

$$DM = \frac{dm}{n(zona_{dm})} \quad (7.10)$$

donde dm es el número total de detecciones múltiples y $n(zona_{dm})$ es el número total de puntos situados dentro de la zona de detección múltiple.

- Localización: para cada punto del borde original, se calcula la localización como

$$loc(x) = \min\{d(x, D), c\} \quad (7.11)$$

donde $d(x, D)$ es la mínima distancia de x a un punto del borde detectado D y $c > 0$ es una constante preestablecida. La constante c establece una cota superior del error. Si un punto del borde ha sido detectado en la imagen a evaluar, entonces $loc(x) = 0$.

La medida de error de la localización se define como:

$$LOC = \frac{1}{n(I) * c} \sum_{x \in I} loc(x) \quad (7.12)$$

donde I representa a los puntos del borde ideal y $n(I)$ es el número total puntos de I . El resultado se normaliza entre 0 y 1 para que todas las medidas de error oscilen en la misma escala.

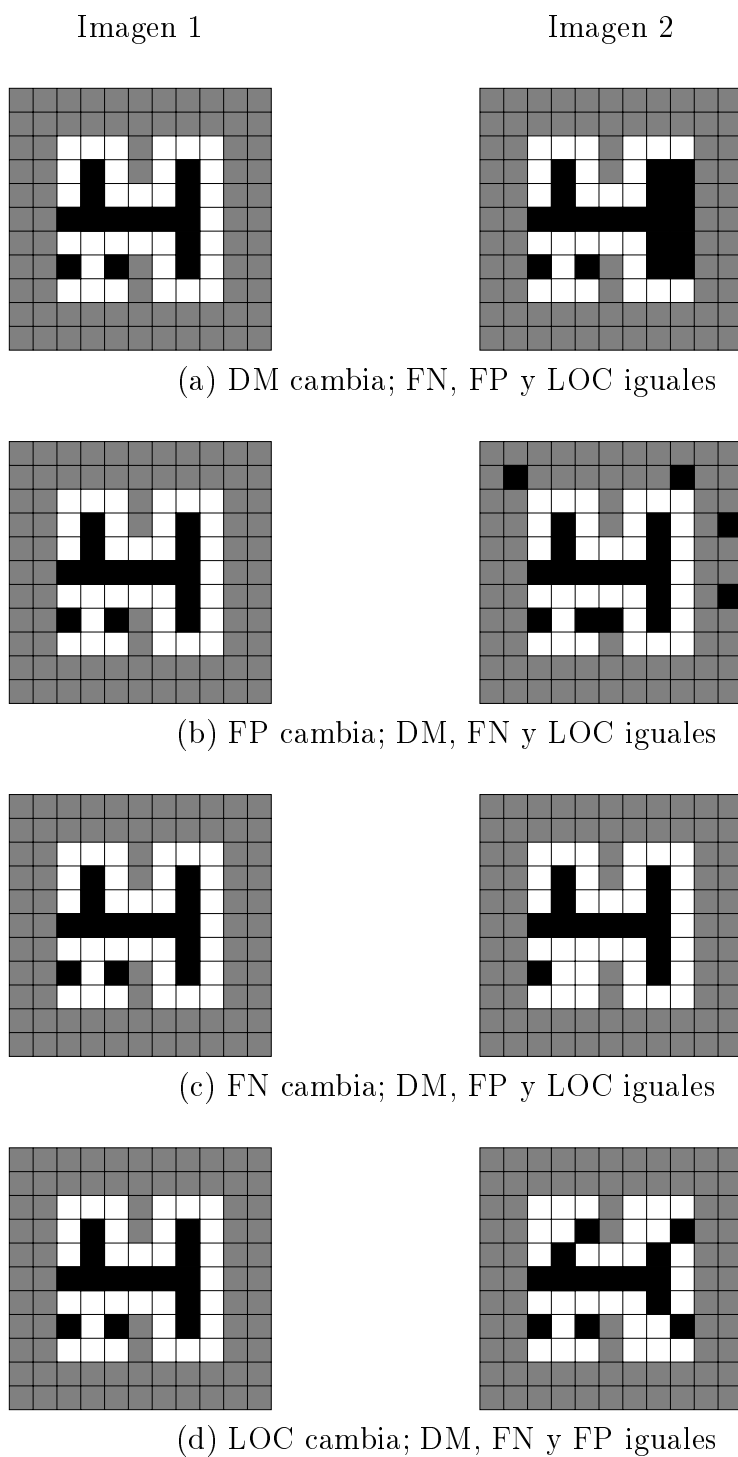
En las pruebas realizadas, se utilizó el valor $c = 5$.

Al igual que el método original de Venkatesh y Kitchen, las nuevas medidas de error definidas también son independientes entre sí, como se puede observar en la figura 7.5.

La medida de fracaso (*failure measure*, FM) se define como:

$$FM = \alpha FP + \beta FN + \gamma DM + \delta LOC \quad (7.13)$$

¹Zona detección = zona ideal \cup zona de detección múltiple.



- Puntos detectados dentro y fuera de la “zona de detección”
- Área fuera de la “zona de detección”

Figura 7.5: Independencia de los tipos de errores: en cada caso sólo cambia la medida de un tipo de error, permaneciendo las demás iguales.

donde

$$\alpha + \beta + \gamma + \delta = 1.0 \quad (7.14)$$

y

$$\alpha, \beta, \gamma, \delta \geq 0.0 \quad (7.15)$$

Tabla 7.1: Coeficientes de las medidas de los tipos de error utilizados en la variante general de Venkatesh y Kitchen.

FM	α	β	γ	δ
1	0'4	0'4	0'1	0'1
2	0'6	0'2	0'05	0'15

Se han definido dos medidas de fracaso utilizando los valores de los parámetros que figuran en la tabla 7.1. En principio, la medida $FM2$ penaliza más a las imágenes con falsos negativos que la medida $FM1$.

7.4.3 Normalización de las medidas de evaluación

Para poder comparar las distintas medidas entre sí, se han normalizado todas las medidas entre 0 (valor óptimo) y 1 (valor pésimo):

- Discrepancia:

$$\tilde{D} = \frac{D}{100.0} \quad (7.16)$$

donde D es la medida definida en la ecuación 7.1.

- Medida de Pratt:

$$\tilde{F} = 1 - F \quad (7.17)$$

donde F es la variante de la medida de Pratt definida en la ecuación 7.6.

- Medida de Baddeley:

$$\tilde{\Delta} = \frac{\Delta_w^2}{c} \quad (7.18)$$

donde Δ_w^2 es la medida de Baddeley definida en la ecuación 7.4, $c = 5$ y N es el número total de puntos de la imagen.

Las medidas de fracaso de la variante de la medida de Venkatesh y Kitchen (ecuación 7.13) ya están normalizadas entre 0 y 1.

7.5 Evaluación de las medidas de calidad

Se han comparado las siguientes medidas de calidad:

- Porcentaje de discrepancia normalizado.
- Versión general y normalizada de la medida de Pratt ($\alpha = 1$ y $\alpha = \frac{1}{9}$).
- Medida normalizada de Baddeley.
- Dos medidas de fracaso (tabla 7.1) de la versión general y normalizada de la medida de Venkatesh y Kitchen.

El proceso de evaluación de la medida de discrepancia ha sido el siguiente:

1. Diseño de las imágenes artificiales en color.
2. Generación de imágenes artificiales con diferentes niveles de ruido exponencial o gaussiano. La varianza del ruido fue $\sigma^2 \in \{1, 3, 10\}$, intentando reflejar situaciones con niveles de ruido bajo, moderado y alto.
3. Aplicación de los algoritmos de detección de bordes en color a las imágenes artificiales sin ruido y con diferentes niveles de ruido exponencial o gaussiano.
4. Umbralización porcentual de cada una de las imágenes generadas por los detectores de bordes. Cada imagen fue umbralizada utilizando un porcentaje que variaba entre 0% y 30% con incrementos del 1%.
5. Uso de las medidas de calidad para obtener la imagen umbralizada cuyo valor sea óptimo, así como para generar la curva de calidad asociada a cada imagen producida por cada detector de bordes. La curva de calidad asocia una medida de error a cada una de las imágenes umbralizadas de cada detector.
6. Análisis de las imágenes propuestas como óptimas para cada detector (figuras 7.8, 7.10, 7.12) y de las curvas de calidad (figuras 7.9 y 7.11).

7.5.1 Diseño de las imágenes artificiales

Se han generado ocho imágenes artificiales en color (figura 7.6):

- Cuatro imágenes con bordes verticales (“barras 90”), horizontales (“barras 0”) e inclinados (“barras 30 y “barras 45”).

- Dos imágenes con bordes curvos, denominadas “círculos” y “elipses”.
- Imagen denominada “barras30-elipses” obtenida a partir de la media aritmética de las imágenes barras 30 y elipses.
- Imagen denominada “barras0-barras90” obtenida a partir de la media aritmética de las imágenes barras 0 y barras 90.

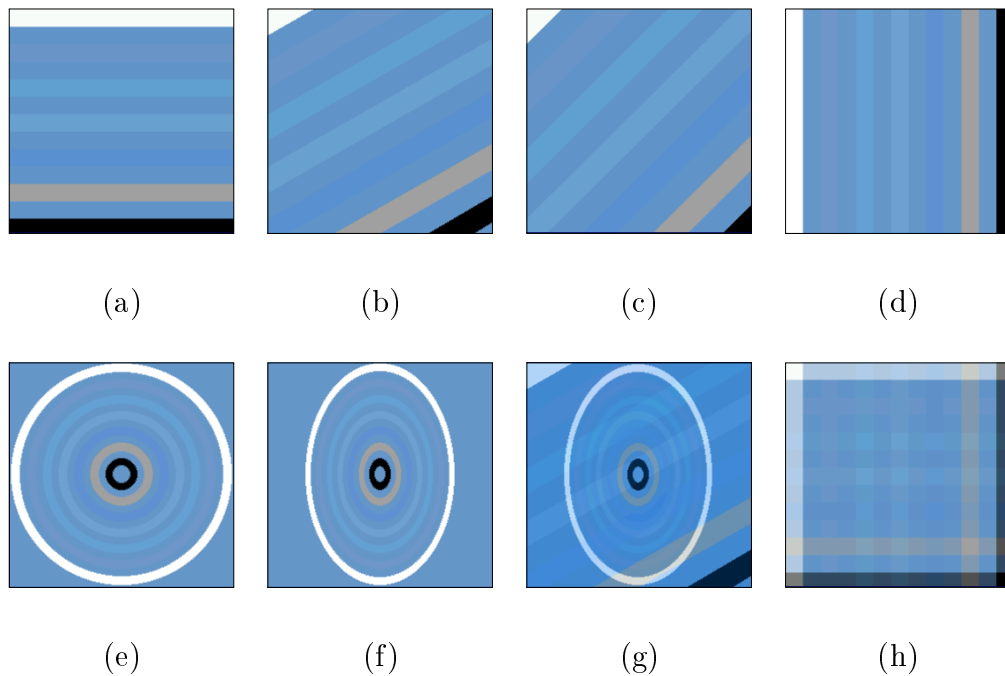


Figura 7.6: Imágenes artificiales en color diseñadas: (a) barras 0, (b) barras 30, (c) barras 45, (d) barras 90, (e) círculos, (f) elipses, (g) barras 30 - elipses y (h) barras 0 - barras 90.

Las características de las imágenes generadas fueron:

- Espacio de color: *RGB*.
- 256 niveles de intensidad ($[0..255]$).
- Tamaño: 256×256 puntos.
- Los bordes provocados por (Trahanias y Venetsanopoulos 1996; Zhu et al. 1999):
 - Cambios en una, dos o tres componentes de color.
 - Cambios en las tres componentes de color, pero manteniendo constante la intensidad.
 - La aparición de zonas blancas, negras o grises.

- Todos los bordes generados eran del tipo de “rampa” ligeramente asimétrica.
- Los bordes generados eran rectos o curvos.
- Se obtuvieron variantes de las imágenes mediante la adición de ruido gaussiano ($\sigma^2 \in \{1, 3, 10\}$) y de ruido exponencial ($\sigma^2 \in \{1, 3, 10\}$).

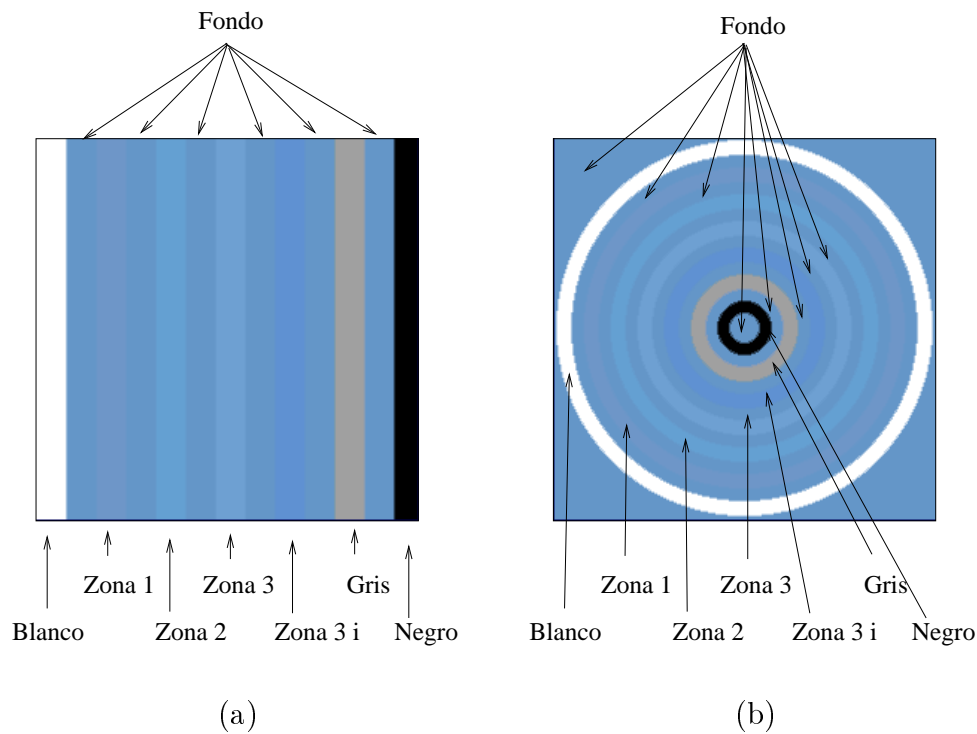


Figura 7.7: Zonas de las imágenes artificiales en color: (a) barras 90 y (b) círculos.

La figura 7.7 muestra las diferentes zonas en las que fueron divididas las imágenes barras 90 y círculos. La tabla 7.2 muestra los valores de las componentes de color de cada una de dichas zonas. Se ha de indicar que la “zona 1” provoca un borde debido a un cambio en el plano rojo, que la “zona 2” provoca un borde al cambiar los planos verde y azul y que la “zona 3” provoca un borde al cambiar los tres planos de color. La “zona 3 i” provoca un borde al cambiar los tres planos de color pero manteniendo constante la intensidad.

7.5.2 Algoritmos de detección de bordes utilizados

Para seleccionar la medida de calidad, se utilizaron los algoritmos de detección de bordes en color que posteriormente fueron evaluados (capítulo 8):

- Algoritmos basados en la noción de gradiente multidimensional: detectores de Di Zenzo (1986) y Lee-Cok (1991).

Tabla 7.2: Valores de las componentes de color de cada una de las zonas de las imágenes artificiales.

Zona	Componentes de color		
	Rojo	Verde	Azul
Fondo	100	150	200
Fondo - Blanco	193	213	233
Blanco	255	255	255
Blanco - Fondo	162	192	222
Fondo - Zona 1	104	150	200
Zona 1	110	150	200
Zona 1 - Fondo	106	150	200
Fondo - Zona 2	100	154	204
Zona 2	100	160	210
Zona 2 - Fondo	100	156	206
Fondo - Zona 3	104	154	204
Zona 3	110	160	210
Zona 3 - Fondo	106	156	206
Fondo - Zona 3 i	97	147	206
Zona 3 i	95	145	210
Zona 3 i - Fondo	98	148	204
Fondo - Gris	124	154	176
Gris	160	160	160
Gris - Fondo	136	156	184
Fondo - Negro	40	60	80
Negro	0	0	0
Negro - Fondo	60	90	120

- Algoritmo basado en operadores vectoriales direccionales: detector de Scharcanski y Venetsanopoulos (1997).
- Algoritmos basados en la diferencia de vectores: detector “Dv” (*difference vector*, vector diferencia) y detector “Dv-hv” (vector diferencia que sólo tiene en cuenta las direcciones horizontal y vertical) (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999).
- Algoritmo basado en el ordenamiento de vectores: detector “Mvd” (*minimum vector dispersion*, vector de dispersión mínima) (Plataniotis y Venetsanopoulos 2000; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999).
- Algoritmo basado en una distancia métrica para estimar el vector gradiente: detector “Máximo de las distancias euclídeas” (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999).

- Algoritmo basado en la entropía: detector de Shiozaki (1986) .
- Algoritmo basado en el uso de histogramas: detector de Pietikäinen y Harwood (1986).
- Algoritmo basado en la suma de las magnitudes de los gradientes monocromáticos de cada una de las componentes de color: detector “Suma de las magnitudes de los gradientes” (Hedley y Yan 1992; Novak y Shafer 1987; Robinson 1976; Robinson 1977a; Pratt 1991; Zenzo 1986).

Los algoritmos de detección de bordes en color fueron aplicados a las imágenes artificiales sin ruido o con diferentes niveles de ruido exponencial o gaussiano. Las imágenes generadas por cada detector fueron umbralizadas con diferentes porcentajes que variaban entre 0% y 30% con incrementos del 1%.

7.5.3 Análisis de los resultados de las medidas de evaluación

El análisis del rendimiento de las medidas de evaluación ha sido realizado de la siguiente forma:

- Para cada combinación de medida de calidad, detector e imagen, se seleccionó el umbral que generaba la imagen umbralizada “óptima” para dicha combinación.
- Las diferentes imágenes propuestas como “óptimas” por cada medida de calidad fueron examinadas y comparadas entre sí.
- Las curvas de error generadas por cada medida de calidad fueron examinadas para comprobar si permitían localizar sin ambigüedad la imagen “óptima”. Se comprobaba si el umbral donde se alcanzaba el valor “óptimo” estaba claramente definido o si, por el contrario, estaba situado en una zona amplia donde todos los umbrales generaran imágenes cuyos valores de la medida de calidad fueran similares.

A continuación se analiza el comportamiento de cada una de las medidas de evaluación en función de los resultados obtenidos.

El uso del porcentaje de discrepancia para medir la calidad de un detector ha presentado los siguientes inconvenientes:

- En la mayoría de las ocasiones, la imagen que se propone como óptima dista mucho de serlo, sobre todo si se compara con las propuestas por otras medidas. Obsérvense las figuras 7.8, 7.10 y 7.12.

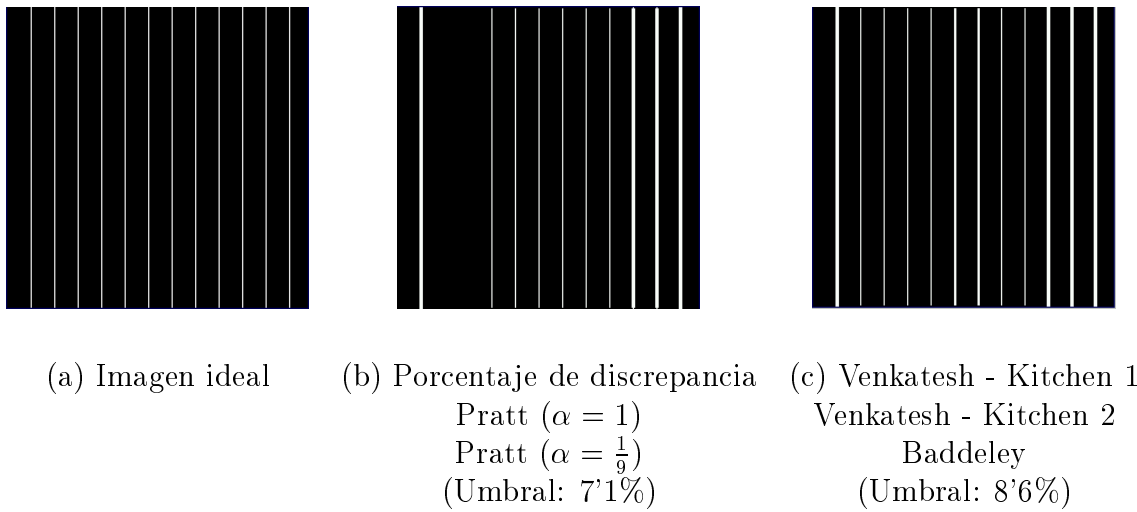


Figura 7.8: Imágenes “óptimas” obtenidas por las diferentes medidas de calidad, al umbralizar la imagen generada al aplicar el detector de Di Zenzo a la imagen barras 90 sin ruido. El porcentaje indica el umbral donde cada medida obtiene el valor óptimo.

- En general, el umbral donde se alcanza el valor óptimo es muy bajo o incluso cero, como ocurrió al aplicar el detector del “Máximo de las distancias” a la imagen de las elipses sin ruido o el detector de Pietikainen a la imagen de las elipses con ruido exponencial de varianza igual a 1.0. Este problema se manifiesta más claramente en imágenes con ruido (figuras 7.10 y 7.12).
- Si se examinan las curvas de error de la discrepancia (figuras 7.9 y 7.11), se observa que:
 - El rango de valores en el que distribuye los valores de las imágenes umbralizadas es muy pequeño en comparación con el utilizado por otras medidas.
 - El umbral donde se alcanza el valor óptimo no se encuentra claramente definido, sino que se localiza en una zona muy amplia.
 - Algunas veces aparecen “óptimos locales” correspondientes a umbrales porcentuales muy alejados entre sí. Este hecho es más perceptible en imágenes si ruido (figura 7.9).

La medida de Pratt presenta las siguientes características:

- Muchas veces, el umbral óptimo propuesto no se corresponde con el mejor valor posible, incluso cuando se analizan imágenes sin ruido, como se puede observar en la figura 7.8.
- La elección del parámetro α tiene una gran influencia en el rendimiento de la medida de Pratt, ya que para ciertas imágenes el valor obtenido para $\alpha = 1$

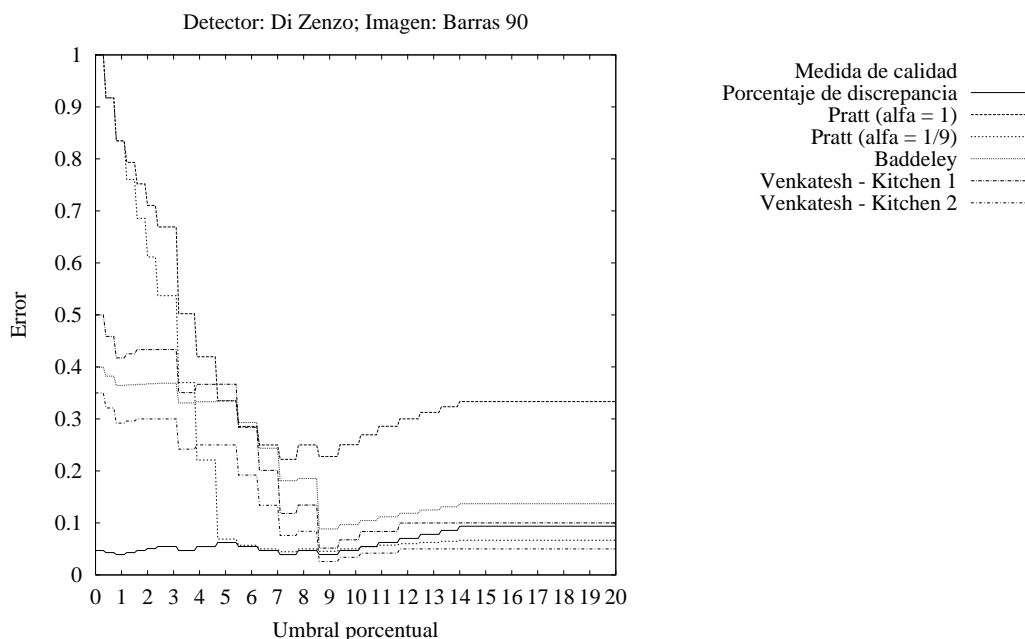


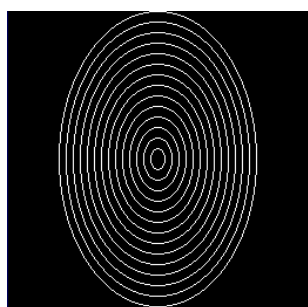
Figura 7.9: Curvas de error de la medidas de calidad correspondientes a la imagen generada por el detector de Di Zenzo al ser aplicado a la imagen barras 90 sin ruido.

es muy diferente del obtenido cuando $\alpha = \frac{1}{9}$. En general, el rendimiento de la medida de Pratt ha mejorado cuando se ha utilizado $\alpha = 1$ (figuras 7.10 y 7.12).

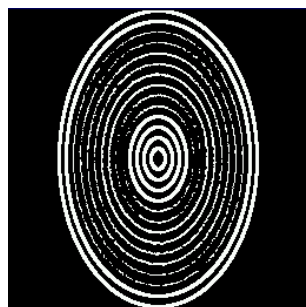
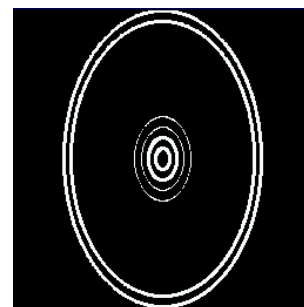
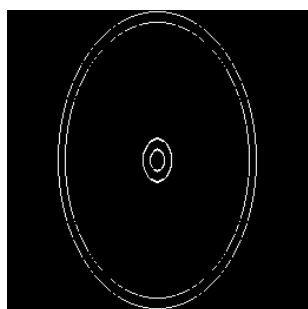
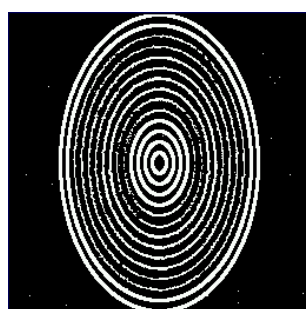
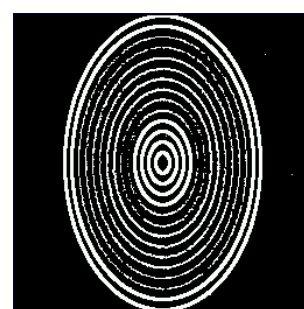
- Si se examinan las curvas de error, se observa que
 - La identificación de la imagen óptima no es clara, puesto que el umbral donde se alcanza el valor óptimo se localiza dentro de una zona muy amplia donde los valores de calidad son muy similares (figuras 7.9 y 7.11).
 - Pueden aparecer múltiples valores óptimos muy diferentes entre sí (figuras 7.9 y 7.11).

Las dos medidas de Venkatesh y Kitchen evaluadas (tabla 7.1) poseen características similares:

- Para imágenes sin ruido o con niveles bajo de ruido, tienen un rendimiento similar al de la medida de Baddeley (figuras 7.8).
- Para imágenes con ruido, favorecen la aparición de falsos positivos (figuras 7.10 y 7.12).



(a) Imagen ideal

(b) Pratt ($\alpha = 1$): 15'50%(c) Pratt ($\alpha = \frac{1}{9}$): 7'00%(d) Porcentaje de
discrepancia: 1'8%(e) Venkatesh - Kitchen 1: 18'20%
Venkatesh - Kitchen 2

(f) Baddeley: 16'60%

Figura 7.10: Imágenes “óptimas” obtenidas por las diferentes medidas de calidad, al umbralizar la imagen generada al aplicar el detector de Scharcanski a la imagen elipses con ruido gaussiano ($\sigma^2 = 3.0$). El porcentaje indica el umbral donde cada medida obtiene el valor óptimo.

- Si se examinan las curvas de error, se puede observar claramente que la localización del valor óptimo es muy deficiente, ya que se encuentra situado en una zona muy amplia donde los valores son muy similares (figura 7.11).

La medida de Baddeley ha obtenido los mejores resultados en las pruebas realizadas, ya que:

- Siempre ha propuesto como imágenes “óptimas” (figuras 7.8, 7.10 y 7.12) a imágenes que son cualitativamente iguales o superiores que las propuestas por las otras medidas.
- Su rendimiento no se ve influenciado por la presencia de ruido, como le ocurre a las otras medidas de calidad (figuras 7.10 y 7.12).
- Si se examinan sus curvas de calidad (figuras 7.9, y 7.11), se puede observar que:

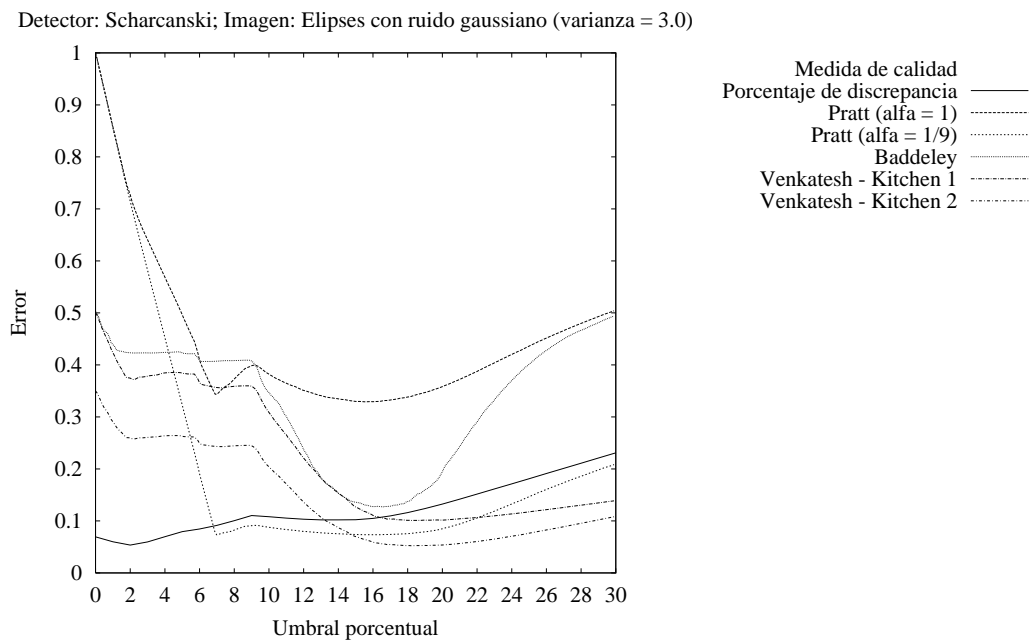
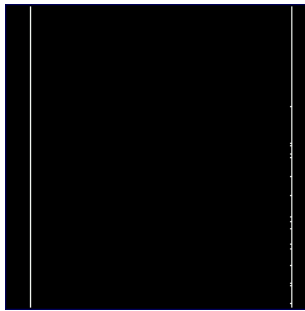


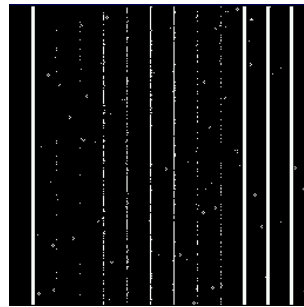
Figura 7.11: Curvas de error de las medidas de calidad correspondientes a la imagen generada por el detector de Scharcanski al ser aplicado a la imagen elipses con ruido gaussiano ($\sigma^2 = 3.0$).

- El umbral propuesto como valor óptimo está siempre bien localizado, a diferencia de las otras medidas de calidad.
- No aparecen óptimos locales distantes entre sí, como le ocurre a la medida de Pratt.

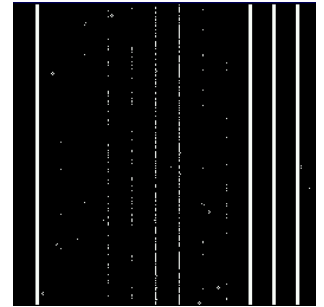
Debido a todos estos motivos, se ha seleccionado la medida propuesta por Baddeley para comparar los algoritmos de detección de bordes en imágenes en color (capítulo 8).



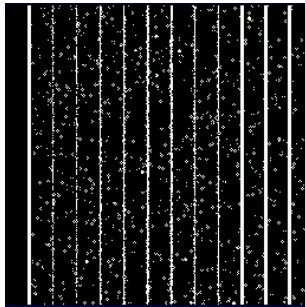
(a) Porcentaje de
discrepancia: 0'8%



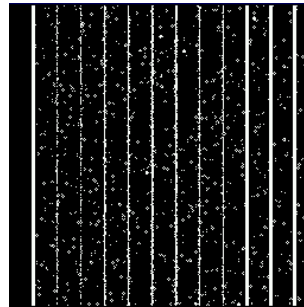
(b) Pratt ($\alpha = 1$): 6'4%



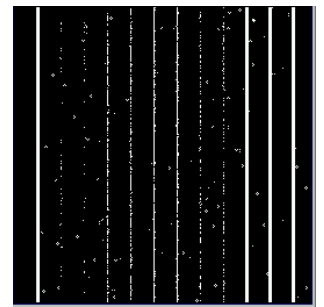
(c) Pratt ($\alpha = \frac{1}{9}$): 5'4%



(d) Venkatesh - Kitchen 1: 11'5 %



(e) Venkatesh - Kitchen 2: 11'0 %



(f) Baddeley: 6'8 %

Figura 7.12: Imágenes “óptimas” obtenidas por las diferentes medidas de calidad al umbralizar la imagen generada al aplicar el detector $D_v - h_v$ a la imagen barras 90 con ruido exponencial ($\sigma^2 = 3.0$). El porcentaje indica el umbral donde se obtiene el valor óptimo.

Capítulo 8

Análisis de los detectores de bordes en color

8.1 Introducción

El sistema de reconocimiento de objetos que se ha diseñado está basado en el uso de un detector de bordes en color. Para elegir el detector más eficiente, se ha utilizado la medida normalizada de Baddeley (1992a,1992b) para evaluar diferentes algoritmos de detección de bordes en color.

La evaluación de los detectores de bordes han consistido en dos pruebas:

- Pruebas con imágenes artificiales sin ruido y con ruido gaussiano o exponencial.
- Pruebas con imágenes reales.

8.2 Algoritmos de detección de bordes en color evaluados

Los algoritmos de detección de bordes en color evaluados han sido:

- Algoritmos basados en la noción de gradiente multidimensional: detectores de Di Zenzo (1986) y Lee y Cok (1991).
- Algoritmo basado en operadores vectoriales direccionales: detector de Scharcanski y Venetsanopoulos (1997).

- Algoritmos basados en la diferencia de vectores: detector “Dv” (*difference vector*, vector diferencia) y detector “Dv-hv” (vector diferencia que sólo tiene en cuenta las direcciones horizontal y vertical) (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999).
- Algoritmo basado en el ordenamiento de vectores: detector “Mvd” (*minimum vector dispersion*, vector de dispersión mínima) (Plataniotis y Venetsanopoulos 2000; Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996; Zhu et al. 1999).
- Algoritmo basado en una distancia métrica para estimar el vector gradiente: detector “Máximo de las distancias euclídeas” (Plataniotis y Venetsanopoulos 2000; Zhu et al. 1999).
- Algoritmo basado en la entropía: detector de Shiozaki (1986).
- Algoritmo basado en el uso de histogramas: detector de Pietikäinen y Harwood (1986).
- Algoritmo basado en la suma de las magnitudes de los gradientes monocromáticos de cada una de las componentes de color: detector “Suma de las magnitudes de los gradientes” (Hedley y Yan 1992; Novak y Shafer 1987; Robinson 1976; Robinson 1977a; Pratt 1991; Zenzo 1986).

Además se ha evaluado el detector de Sobel (González y Woods 1996; Pratt 1991; Sobel 1970; Sonka et al. 1993) para contrastar el rendimiento de un algoritmo monocromático clásico con el de los algoritmos de detección de bordes en color. En este caso, cada imagen en color ha sido transformada en una imagen monocromática mediante el cálculo de la media aritmética de sus componentes (Koschan 1995):

$$I(x, y) = \frac{1}{3}(R(x, y) + G(x, y) + B(x, y)) \quad (8.1)$$

donde $R(x, y)$, $G(x, y)$ y $B(x, y)$ son las componentes de color roja, verde y azul en el punto (x, y) , respectivamente.

Todos estos algoritmos de detección de bordes requieren una fase posterior de umbralización para identificar los puntos que pertenecen a los bordes. Puesto que ninguno de ellos indica qué método de umbralización se ha de aplicar, se ha optado por utilizar el método de umbralización porcentual (Pratt 1978; Sonka et al. 1993).

Con la finalidad de que todos los algoritmos sean evaluados de forma imparcial, se han establecido las siguientes restricciones:

- La ventana de procesamiento utilizada ha sido de tamaño 3×3 , aunque haya algoritmos que puedan ser aplicados con ventanas mayores.

- La métrica utilizada ha sido la distancia euclídea.
- La estimación del gradiente en las componentes de color se ha calculado mediante el operador de Sobel.

El algoritmo “Mvd” posee dos parámetros k y l , que permiten atenuar la influencia del ruido impulsivo y gaussiano, respectivamente. Los autores de este algoritmo han indicado que los valores de k y l han de ser obtenidos experimentalmente (Trahanias y Venetsanopoulos 1993; Trahanias y Venetsanopoulos 1996). En las pruebas realizadas con imágenes artificiales, los valores utilizados fueron $k = 4$ y $l = 4$, mientras que en las pruebas con imágenes reales los valores establecidos fueron $k = 3$ y $l = 3$. En las pruebas realizadas, se ha comprobado que el algoritmo es más sensible respecto de k que de l .

8.3 Pruebas realizadas con imágenes artificiales

Los detectores de bordes en color fueron aplicados a las imágenes artificiales sin ruido y con ruido gaussiano o exponencial ($\sigma^2 \in \{1, 3, 10\}$). Las características de estas imágenes han sido descritas en la subsección 7.5.1 y se pueden observar en la figura 7.6.

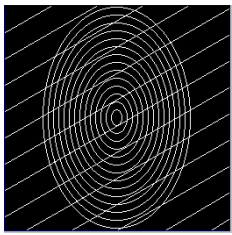
Una vez que se ha aplicado un detector de bordes a una imagen artificial, se ha hecho uso de la medida normalizada de Baddeley (subsección 7.4.3) para determinar el umbral que se ha de utilizar para generar la imagen umbralizada óptima para dicho detector (figuras 8.1 y 8.3).

Los resultados de la medida normalizada de Baddeley obtenidos por cada detector de bordes han permitido generar la figura 8.2, que muestra los valores medios obtenidos por cada uno de los detectores para diferentes niveles de ruido.

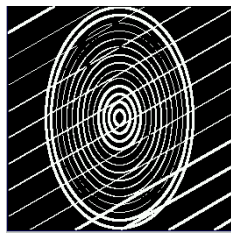
La tabla 8.1 muestra los subconjuntos homogéneos contruidos según el test de Scheffé. Se puede interpretar esta tabla como una clasificación global de todos los detectores.

El análisis de la figura 8.2 y de la tabla 8.1 permite extraer los siguientes resultados de los detectores de bordes, ordenados según su rendimiento global:

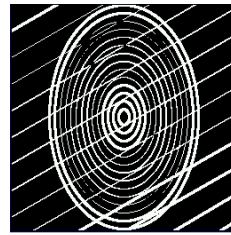
- Los detectores de Di Zenzo y Lee y Cok son los que obtienen los mejores resultados, sobre todo en presencia de ruido y con las imágenes más complejas (barras 0 - barras 90 y barras 30 - elipses) (figuras 8.1 y 8.3).
- Los valores medios obtenidos por los detectores de Di Zenzo y de Lee y Cok son coincidentes. Ello se debe a que el método de Lee y Cok es una variante del método de Di Zenzo.



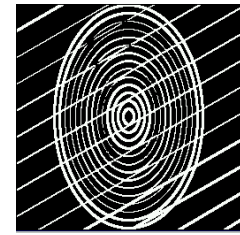
(a) Imagen ideal



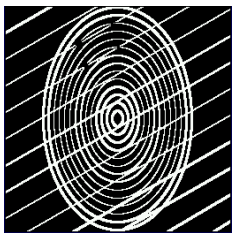
(b) Di Zenzo: 0'1331



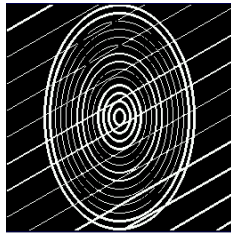
(c) Lee y Cok: 0'1331



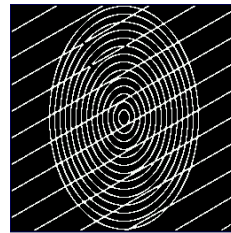
(d) Scharcanski: 0'1383



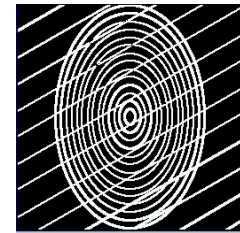
(e) Dv: 0'1431



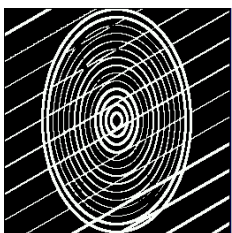
(f) Dv-hv: 0'1015



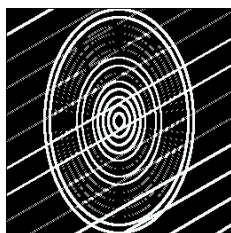
(g) Mvd: 0'0746



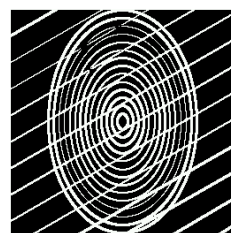
(h) Máximo: 0'1614



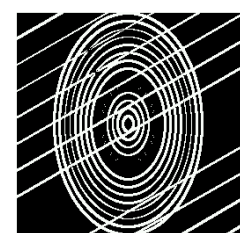
(i) Shiozaki: 0'1364



(j) Pietikäinen: 0'1744



(k) Suma: 0'2119



(l) Sobel: 0'2607

Figura 8.1: Resultados obtenidos por los detectores de bordes al ser aplicados a la imagen barras 30 - elipses sin ruido. En cada caso se muestra la imagen umbralizada y el valor de la medida de Baddeley en la que cada detector alcanza el valor óptimo.

- Para imágenes sin ruido o con niveles bajos de ruido, el detector “Mvd” obtiene los mejores resultados (figuras 8.1). Sin embargo, para valores de ruido moderados o altos, su rendimiento se reduce apreciablemente (figura 8.3). En tales situaciones sería necesario modificar los valores de sus parámetros k y l para mejorar su rendimiento.
- El detector de Scharcanski y Venetsanopoulos tiene un rendimiento similar pero ligeramente peor que el de los detectores de Di Zenzo y Lee y Cok.
- Los detectores “Dv” y “Dv-hv” tienen un rendimiento global homogéneo, aunque en ausencia de ruido, o con niveles de ruido bajos, el detector “Dv-hv” obtiene mejores resultados (figura 8.1). Sin embargo, esta situación se invierte a media que el ruido aumenta (figura 8.3). En presencia de ruido

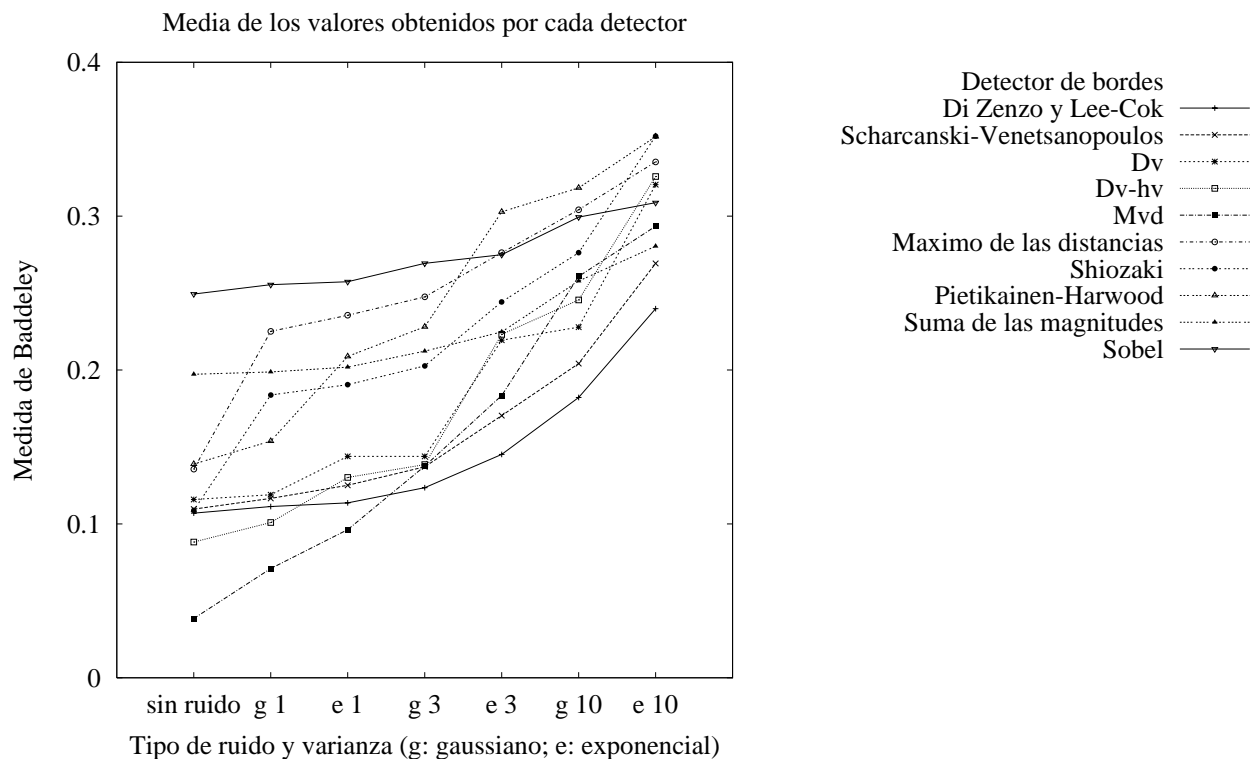


Figura 8.2: Valores medios obtenidos por los detectores de bordes en color al ser aplicados a las imágenes artificiales con diferentes niveles de ruido gaussiano o exponencial.

exponencial elevado ($\sigma^2 = 10$), el rendimiento de ambos detectores llega ser peor que el obtenido por el detector de Sobel.

- En términos globales, los detectores de Shiozaki y de “Suma de magnitudes de los gradientes” pertenecen al mismo grupo homogéneo, pero su rendimiento difiere bastante en ausencia o en presencia de ruido elevado (figuras 8.1 y 8.3).
- El detector de Shiozaki obtiene buenos resultados con imágenes sin ruido (figura 8.1), pero su rendimiento se ve afectado considerablemente por la presencia de ruido, sobre todo en zonas donde la intensidad es baja (figura 8.3).
- El detector de la “Suma de las magnitudes de los gradientes” tiene un rendimiento muy discreto, pero siempre superior al obtenido por el detector de Sobel.
- Los detectores del “Máximo de las distancias” y de Pietikäinen y Harwood

Tabla 8.1: Rendimiento de los detectores de bordes en color con imágenes artificiales: subconjuntos homogéneos según el test de Scheffé. Se muestran las medias para los grupos en subconjuntos homogéneos.

Detector	Subconjunto					
	1	2	3	4	5	6
Lee y Cok	0'1461					
Di Zenzo	0'1461					
Mvd	0'1544	0'1544				
Scharcanski		0'1618				
Dv-hv			0'1789			
Dv			0'1843			
Shiozaki				0'2226		
Suma de magnitudes				0'2247		
Pietikäinen					0'2432	
Máximo de distancias					0'2514	
Sobel						0'2735
Significación	0'687	0'837	0'978	1'000	0'718	1'000

tienen, en general, un rendimiento muy deficiente. Incluso son superados por el detector de Sobel en imágenes con niveles de ruido bajos (figura 8.3).

- El detector de Sobel tiene un rendimiento bajo debido a que no puede detectar los bordes en los que no hay cambios globales de intensidad (figuras 8.1 y 8.3). No obstante, posee un mejor rendimiento en presencia de ruido que algunos detectores de bordes en color.

Las pruebas con imágenes artificiales han puesto de manifiesto que los detectores de Di Zenzo y Lee y Cok son los que obtienen mejores resultados, con la ventaja adicional de no necesitar ningún parámetro.

El detector “Mvd” también obtiene buenos resultados en imágenes sin ruido o con niveles bajos de ruido. Sin embargo, su rendimiento disminuye a medida que el ruido aumenta. Además presenta el inconveniente de que necesita que se ajusten experimentalmente los parámetros k y l . En las pruebas realizadas se utilizaron los valores $k = 4$ y $l = 4$ porque eran los que obtenían unos resultados excelentes en imágenes sin ruido (figuras 8.2 y 8.1). En general, el detector es más sensible al parámetro k que al l .

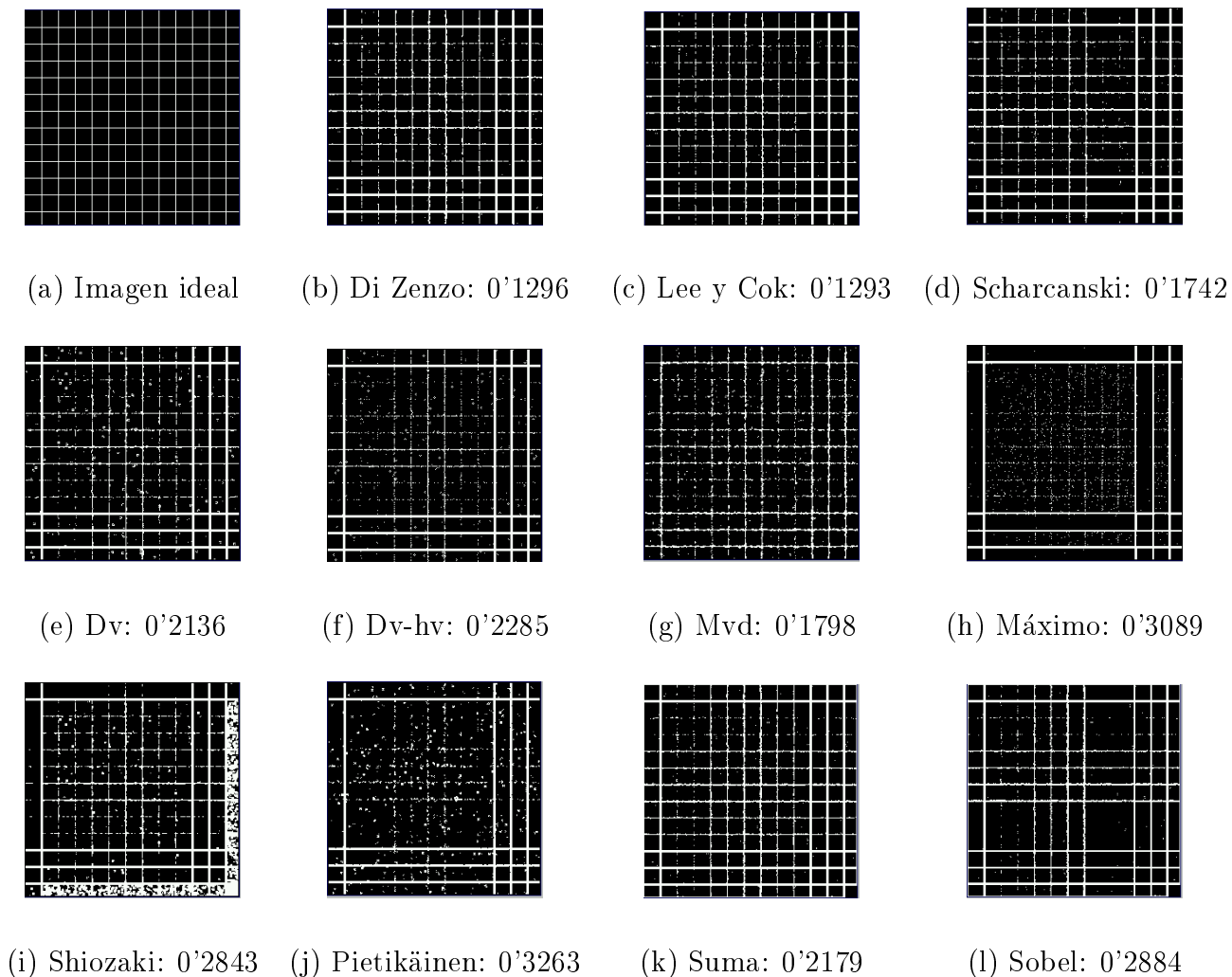


Figura 8.3: Resultados obtenidos por los detectores de bordes al ser aplicados a la imagen barras 0 - barras 90 con ruido exponencial de varianza $\sigma^2 = 1.0$. En cada caso se muestra la imagen umbralizada y el valor de la medida de Baddeley en la que cada detector alcanza el valor óptimo.

8.4 Pruebas realizadas con imágenes reales

La evaluación con imágenes reales presenta la dificultad de carecer de un mapa de bordes ideal. Para superar esta dificultad se ha hecho uso de un método de evaluación relativa basada en el uso de imágenes de consenso.

Bryant y Bouldin (1979) utilizaron una única imagen de consenso para evaluar el rendimiento de los detectores de bordes en imágenes monocromáticas. La imagen de consenso podía ser creada por “votación excluyente o incluyente”. En la votación excluyente, un punto no pertenecía a la imagen de consenso si no había sido detectado por un número mínimo de algoritmos de detección de bordes; en la votación incluyente, el proceso de creación de la imagen de consenso

era el inverso.

En el presente estudio comparativo no se ha utilizado una única imagen de consenso, sino todas las posibles. Dados n detectores de bordes, se han generado n imágenes de consenso, de forma que la imagen de consenso número i ha estado compuesta por aquellos puntos que son considerados como puntos de bordes por “al menos” i algoritmos de detección. Cada una de las imágenes de consenso ha sido utilizada como mapa de bordes ideal para evaluar los detectores de bordes en color.

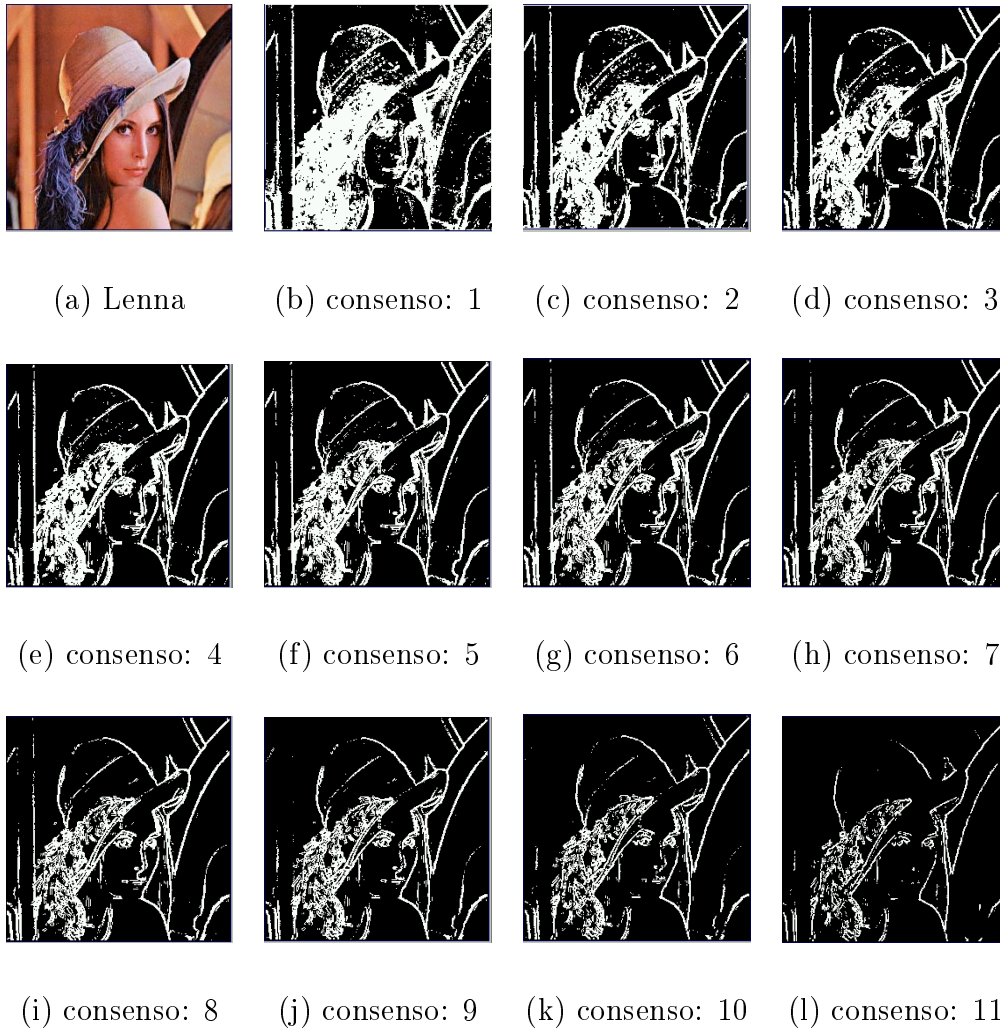


Figura 8.4: Imágenes de consenso obtenidas al utilizar un umbral del 20% en las imágenes generadas por los detectores de bordes al ser aplicados a la imagen en color Lenna (figura 8.5). La imagen de nivel de consenso i está compuesta por aquellos puntos que han sido considerados como puntos de borde por “al menos” i detectores.

Sin embargo, para obtener los puntos de borde de cada una de las imágenes generadas por cada detector es necesario umbralizar dichas imágenes. Puesto que ningún algoritmo de detección de bordes indica cuál debe ser el método

de umbralización que se ha de utilizar, se ha optado por aplicar el método de umbralización porcentual.

La figura 8.4 muestra las imágenes de consenso correspondientes a la imagen Lenna. Estas imágenes se han obtenido umbralizando al 20% las imágenes generadas por los detectores de bordes, las cuales se muestran en la figura 8.5.

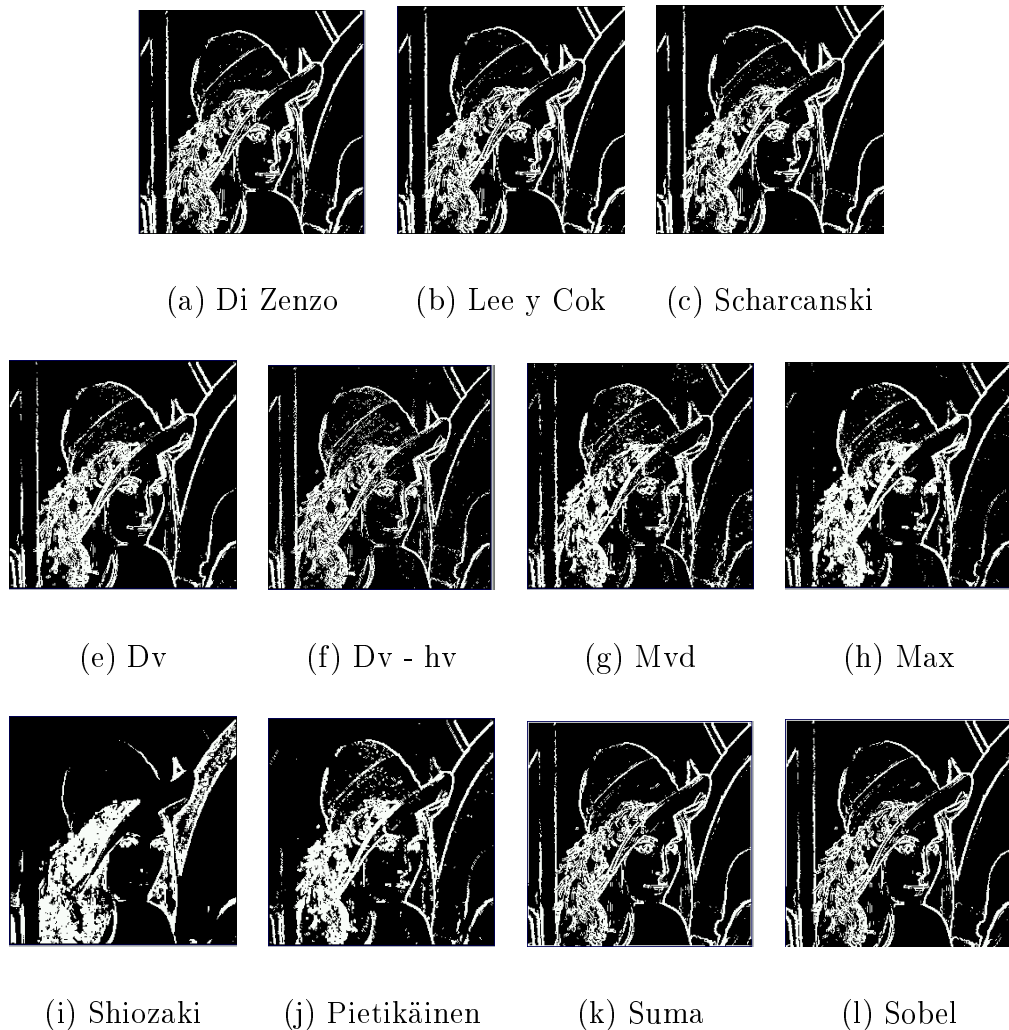


Figura 8.5: Imágenes obtenidas al utilizar un umbral del 20% en las imágenes generadas por los detectores de bordes al ser aplicados a la imagen en color Lenna.

En resumen, el procedimiento utilizado para efectuar la evaluación de los detectores de bordes ha sido el siguiente:

- Aplicación de los detectores de bordes a las imágenes en color.
- Umbralización de las imágenes generadas por los detectores.
- Generación de las imágenes de consenso.

- Uso de la medida de Baddeley para comparar la imagen umbralizada de cada detector con cada una de las imágenes de consenso generadas.

Este procedimiento se ha repetido para tres umbrales porcentuales (10%, 15% y 20%) y se han utilizado doce imágenes reales en color muy populares¹.

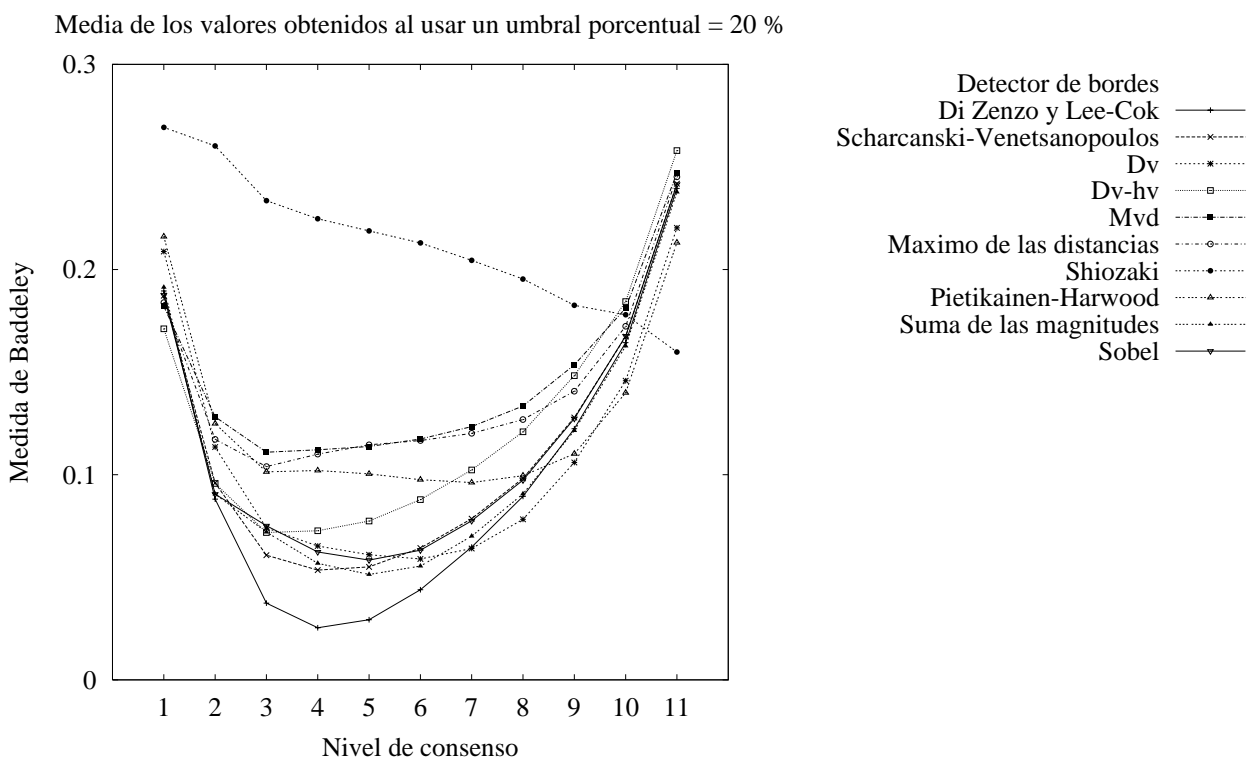


Figura 8.6: Valores medios obtenidos por los detectores de bordes en color al ser aplicados a las imágenes reales y utilizar un umbral del 20%.

La figura 8.6 muestra los valores medios de la medida normalizada de Baddeley obtenidos por los detectores de bordes al utilizar un umbral porcentual del 20%.

El análisis de las imágenes de consenso de la figura 8.4, de las curvas de la gráfica 8.6 y de la tabla 8.2 permiten deducir los siguientes resultados:

- No todas las imágenes de consenso son adecuadas para efectuar una evaluación justa. Las imágenes de consenso con un número de consenso bajo

¹Imágenes utilizadas: Amber, Blocks, Car, Cheetah, Fruit, Gull, House, Lenna, Lisa, Mandrill, Saturno y Valve.

Tabla 8.2: Imágenes de consenso: subconjuntos homogéneos según el test de Scheffé. La tabla muestra las medias para los grupos en subconjuntos homogéneos teniendo en cuenta todos los umbrales porcentuales utilizados (10%, 15% y 20%).

Imagen de consenso	Subconjunto							
	1	2	3	4	5	6	7	8
5	0'0740							
4	0'0746							
6	0'0776							
3	0'0802	0'0802						
7		0'0853						
8			0'0972					
2				0'1088				
9					0'1164			
10						0'1441		
1							0'1929	
11								0'2017
Significación	0'087	0'315	1'000	1'000	1'000	1'000	1'000	1'000

introducen demasiado ruido, mientras que las que exigen un consenso muy alto carecen de puntos que deben pertenecer al mapa de bordes ideal.

- Además, debido a las similitudes en la definición de diferentes detectores², el nivel de consenso debe ser superior al número de algoritmos que pertenezcan a una misma clase. En nuestro caso, el nivel de consenso mínimo ha de ser 3.
- Si se examinan los subconjuntos homogéneos creados según el test de Scheffé (tabla 8.2), se observa que las imágenes de consenso comprendidas entre 3 y 7 forman dos subconjuntos homogéneos, mientras que el resto de imágenes de consenso forman subconjuntos unitarios con valores medios muy diferentes entre sí.

Como consencuencia de lo anterior, el rendimiento de los detectores de bordes con imágenes reales sólo se ha realizado teniendo en cuenta los resultados obtenidos con imágenes de consenso comprendidas entre 3 y 7.

El análisis de las imágenes generadas por los detectores de bordes (como las de las figura 8.5), de las gráficas de evaluación relativa correspondientes a los umbrales utilizados (como la gráfica 8.6) y de la tabla de subconjuntos homogéneos

²Por ejemplo, los algoritmos de Di Zenzo y Lee y Cok son muy similares entre sí, al igual que los detectores “Dv” y “Dv-hv”.

creada según el test de Scheffé para los detectores de bordes en color (tabla 8.3) refleja que:

- Los detectores de bordes de Di Zenzo y Lee y Cok tienen un rendimiento claramente superior al de los demás detectores de bordes.
- Los detectores de Scharcanski y Venetsanopoulos, “Dv” y “Suma de las magnitudes de los gradientes” tienen un rendimiento similar.
- El detector monocromático de Sobel supera a otros detectores de bordes en color.
- Los detectores “Dv-hv”, de Pietikäinen y Harwood y del “Máximo de las distancias” tienen un rendimiento global inferior al detector de Sobel.
- El detector “Mvd” ha tenido un rendimiento muy deficiente, debido a una mala elección de los parámetros. Probablemente, una elección experimental de los parámetros permitiría mejorar su rendimiento. Sin embargo, la dependencia de los dos parámetros provoca que el rendimiento de este detector de bordes sea muy poco robusto.
- El detector de Shiozaki presenta unos resultados extraordinariamente pobres debido a su alta sensibilidad al ruido.

Tabla 8.3: Rendimiento de los detectores de bordes en color con imágenes reales utilizando sólo imágenes de consenso comprendidas entre 3 y 7 y todos los umbrales porcentuales utilizados (10%, 15% y 20%). Se muestran los valores medios de los subconjuntos homogéneos según el test de Scheffé.

Detector	Subconjunto						
	1	2	3	4	5	6	7
Lee y Cok	0'0329						
Di Zenzo	0'0329						
Scharcanski		0'0534					
Dv		0'0536					
Suma de magnitudes		0'0561					
Sobel			0'0618				
Dv-hv				0'0741			
Pietikäinen					0'0881		
Máximo de distancias						0'1022	
Mvd						0'1051	
Shiozaki							0'2018
Significación	1'000	0'677	1'000	1'000	1'000	0'560	1'000

8.5 Análisis de los resultados

La medida normalizada de Baddeley ha sido utilizada para evaluar el rendimiento de varios detectores de bordes en color y del detector de Sobel, para lo cual se han utilizado imágenes en color, tanto artificiales (sin ruido y con ruido gaussiano o exponencial), como reales. Los resultados permiten extraer las siguientes conclusiones:

- Los detectores de Lee y Cok y Di Zenzo son los que han obtenido un mejor rendimiento en todas las pruebas realizadas.
- El detector “Mvd” obtiene un gran rendimiento en imágenes artificiales sin ruido o con bajos niveles de ruido, pero su rendimiento con imágenes con ruido moderado o alto o con imágenes reales ha sido muy deficiente. Una correcta elección de sus parámetros permitiría mejorar su rendimiento. Sin embargo, esta dependencia de los parámetros es su principal inconveniente.
- El detector monocromático de Sobel obtiene un rendimiento superior a algunos detectores de bordes en color cuando se aplica a imágenes reales, pero su rendimiento es muy inferior al obtenido por los detectores de Lee y Cok y Di Zenzo.

Por tanto, las pruebas realizadas han indicado con claridad que los mejores detectores de bordes en color evaluados han sido los algoritmos de Di Zenzo (1986) y de Lee y Cok (1991).

Capítulo 9

Diseño de un algoritmo de adelgazamiento de bordes

9.1 Introducción

Se ha diseñado un algoritmo de adelgazamiento de bordes basado en la versión “paralela” del método propuesto por O’Gorman (1.990)¹. Se ha de indicar que los algoritmos de adelgazamiento se denominan “secuenciales” o “paralelos” dependiendo de cómo se efectúe el procesamiento de los puntos:

- Algoritmo secuencial: el resultado de procesar un punto depende de los valores de los puntos tratados previamente en la misma iteración.
- Algoritmo paralelo: el resultado de procesar un punto depende de los valores de los puntos obtenidos en la iteración anterior. Debido a ello, todos los puntos podrían ser evaluados simultáneamente.

El nuevo algoritmo “paralelo” utiliza ventanas de tamaño 3×3 y posee dos versiones, dependiendo de que se desee utilizar la conectividad de orden 8 o de orden 4.

En la sección 9.2 se explica la versión del nuevo algoritmo que utiliza la conectividad 8, mientras que la versión basada en la conectividad 4 es descrita en la sección 9.3.

¹El algoritmo de O’Gorman es descrito en el capítulo 5.

9.2 Variante basada en la conectividad de orden 8

El algoritmo de O’Gorman (1.990) establece tres criterios para poder borrar un punto marcado que, teniendo en cuenta que se van a utilizar ventanas de tamaño 3×3 , se pueden interpretar del siguiente modo (figura 9.1):

1. Los puntos marcados del perímetro han de estar conectados entre sí (sin utilizar el punto central) mediante conectividad 8.
2. El punto central ha de tener al menos dos vecinos marcados.
3. Al menos debe haber en el perímetro una cadena de puntos no marcados cuya longitud sea superior a 1.

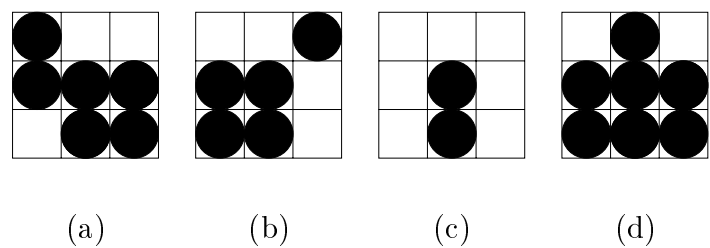


Figura 9.1: Criterios del algoritmo de adelgazamiento con conectividad 8: (a) patrón que cumple los criterios; (b) patrón con puntos marcados del perímetro que no están conectados entre sí mediante conectividad 8; (c) patrón en el que el punto central tiene menos de dos vecinos marcados; (d) patrón en el que la cadena más larga de puntos no marcados del perímetro no es superior a 1.

-1	-2	-4
-128	256	-8
-64	-32	-16

Figura 9.2: Máscara de convolución utilizada por el algoritmo de adelgazamiento.

Se pueden identificar unívocamente los patrones que cumplen los criterios del algoritmo de adelgazamiento si se convoluciona la imagen de entrada con la máscara de la figura 9.2. La figura 9.3 muestra la identificación numérica de los patrones que permiten borrar puntos de la capa norte del contorno. Los

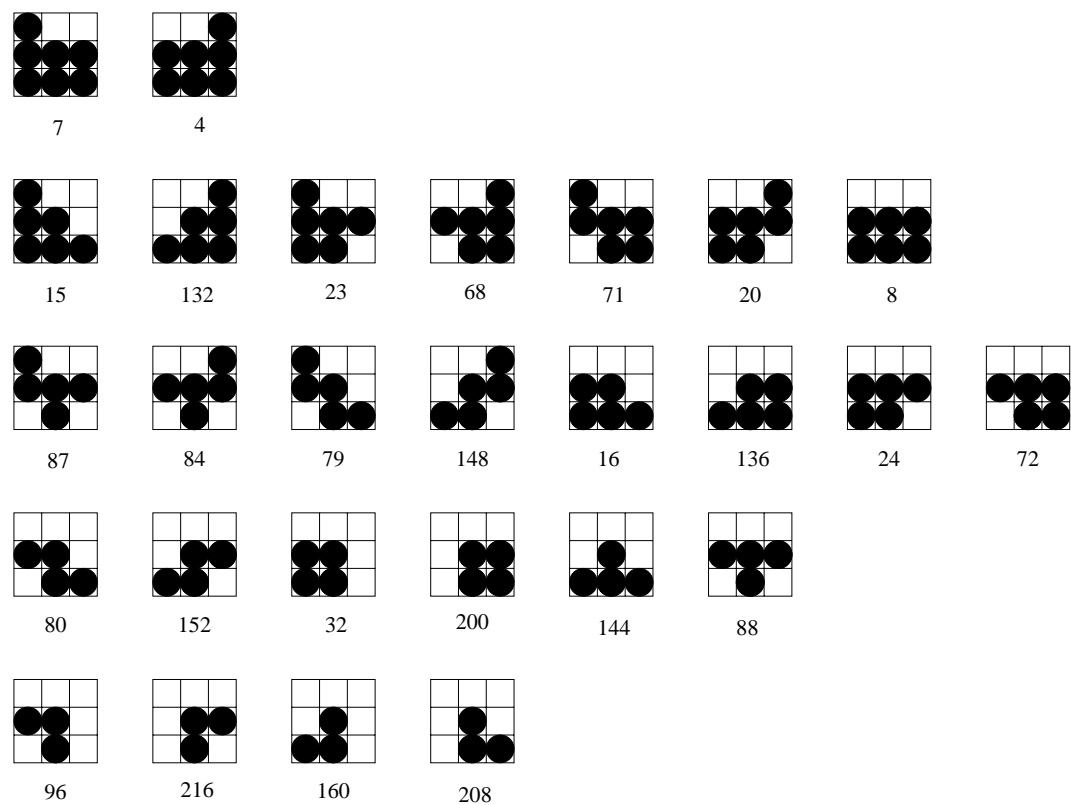


Figura 9.3: Codificación numérica de los patrones que permiten eliminar puntos del lado norte de un contorno respetando la conectividad de orden 8.

patrones de las capas este, sur y oeste se obtienen rotando -90° , -180° y -270° , respectivamente, los patrones de la capa norte.

Se ha considerado que un patrón es de tipo norte (este, sur, oeste) si es similar el patrón “Norte” (“Este”, “Sur”, “Oeste”) de la figura 9.4.

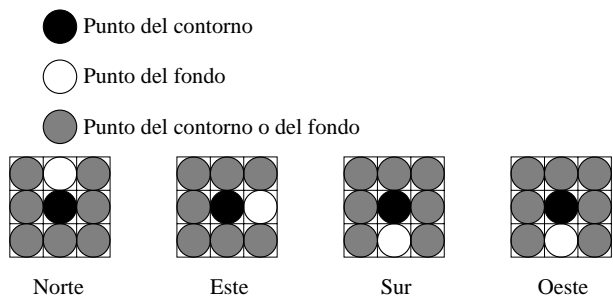


Figura 9.4: Tipos de patrones.

Existen patrones que pueden ser de dos o más tipos. Considérese, por ejemplo, el patrón con valor 15 de la figura 9.3 que puede ser considerado como patrón de tipo norte o este. Sin embargo, para evitar que la aplicación del algoritmo no sea simétrica, también se incluye entre los patrones que eliminan puntos de la capa norte al patrón con valor 132 que es de los tipos norte y oeste y, además,

es simétrico del 15 respecto del eje vertical. Se ha aplicado el mismo criterio en todos los demás casos similares.

El algoritmo de adelgazamiento permite borrar los puntos cuyo vecindario se corresponda con algunos de los patrones que cumplan los requisitos de eliminación. Sin embargo, esta operación no puede ser realizada con todos los patrones de todas las capas simultáneamente, puesto que podría borrarse un número excesivo de puntos. Debido a ello, se han de aplicar iterativamente cuatro fases en las que se eliminarán en primer lugar los puntos del contorno cuyo vecindario se ajuste a los patrones de la capa del lado norte y, a continuación, los correspondientes a los lados sur, este y oeste.

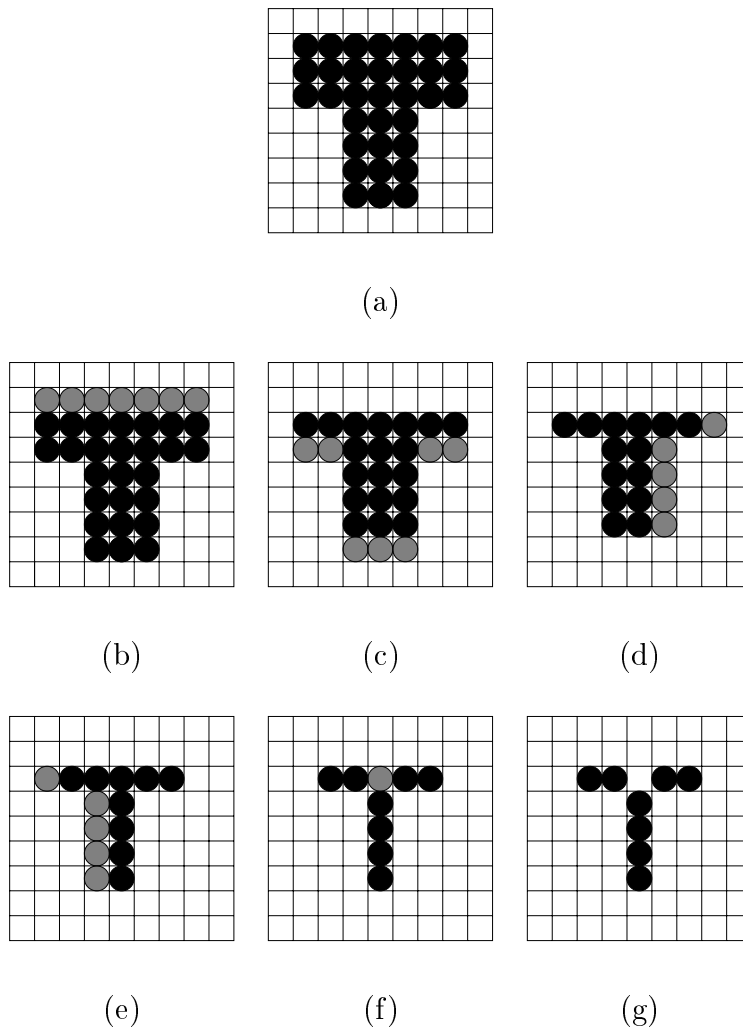


Figura 9.5: Fases del algoritmo de adelgazamiento de bordes con conectividad 8: (a) contorno original; eliminación de puntos de los lados (b) norte (primera vez), (c) sur, (d) este, (e) oeste y (f) norte (segunda vez); (g) contorno adelgazado.

La figura 9.5 muestra cómo se obtendría el contorno adelgazado de un objeto mediante la aplicación sucesiva de cada una de las cuatro fases. El proceso de adelgazamiento termina cuando no es posible eliminar ningún punto en

cualquiera de las cuatro fases.

Puesto que el algoritmo de adelgazamiento se aplica en cuatro fases sucesivas, sólo es necesario analizar “los valores” de los patrones de la capa norte. Obsérvese que, durante la fase norte, se ha de convolucionar la imagen con la máscara de la figura 9.2 para eliminar los puntos que se ajusten a uno de los patrones de la capa norte (figura 9.3) y que, por tanto, obtendrán un valor de convolución perteneciente al siguiente conjunto

$$S_8 = \{ \begin{array}{l} 4, 7, 8, 15, 16, 20, 23, 24, 32, \\ 68, 71, 72, 79, 80, 84, 87, 88, 96, \\ 132, 136, 144, 148, 152, 160, 200, 208, 216 \end{array} \} \tag{9.1}$$

Si durante la fase sur se aplica, en cambio, la máscara de convolución de la figura 9.6(b), entonces los puntos que se ajusten a los patrones de la capa sur obtendrán valores que también pertenecerán al conjunto S , mientras que los patrones de las otras capas obtendrán valores diferentes. Para las fases este y oeste se han de utilizar las máscaras de las figuras 9.6(a) y (c), respectivamente, para obtener resultados similares.

-64	-128	-1
-32	256	-2
-16	-8	-4

(a)

-16	-32	-64
-8	256	-128
-4	-2	-1

(b)

-4	-8	-16
-2	256	-32
-1	-128	-64

(c)

Figura 9.6: Máscaras de convolución utilizadas al eliminar los puntos de la capa (a) este, (b) sur y (c) oeste.

9.3 Variante basada en la conectividad de orden 4

La variante del algoritmo de adelgazamiento basada en la conectividad de orden 4 sustituye el primer criterio de adelgazamiento de la versión anterior por el siguiente:

1. Los puntos marcados del perímetro han de estar conectados entre sí (sin utilizar el punto central) mediante “conectividad 4”, descartándose las conexiones mediante conectividad 8.

Al introducir la restricción de la conectividad de orden 4, han de eliminarse algunos de los patrones de la versión anterior que suprimen puntos de la capa norte (figura 9.3), permitiéndose sólo los patrones que se muestran en la figura 9.7, los cuales se corresponden con los valores del siguiente conjunto:

$$S_4 = \{ 4, 7, 8, 15, 16, 32, 132, 136, 144, 160, 200, 208 \} \quad (9.2)$$

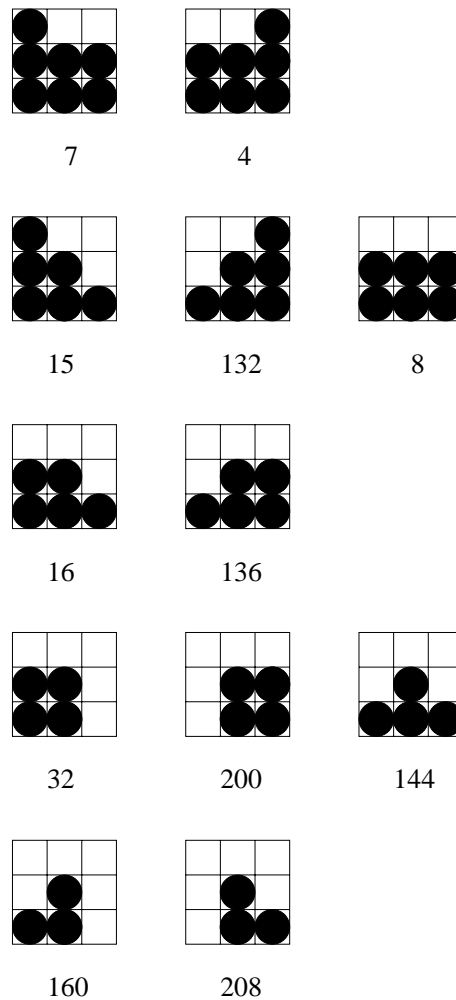


Figura 9.7: Codificación numérica de los patrones que permiten eliminar puntos del lado norte de un contorno respetando la conectividad de orden 4.

Los patrones de las capas este, sur y oeste se obtienen rotando -90° , -180° y -270° , respectivamente, los patrones de la capa norte.

La aplicación de esta versión es similar a la del algoritmo basado en la conectividad de orden 8.

9.4 Aplicación de los algoritmos

La figura 9.8 muestra los esqueletos obtenidos por los algoritmos de adelgazamiento diseñados. Como se puede observar, las diferencias entre ambas versiones se encuentran en las “rampas” de los esqueletos, puesto que el método basado en la conectividad de orden 4 no puede efectuar conexiones diagonales entre puntos.

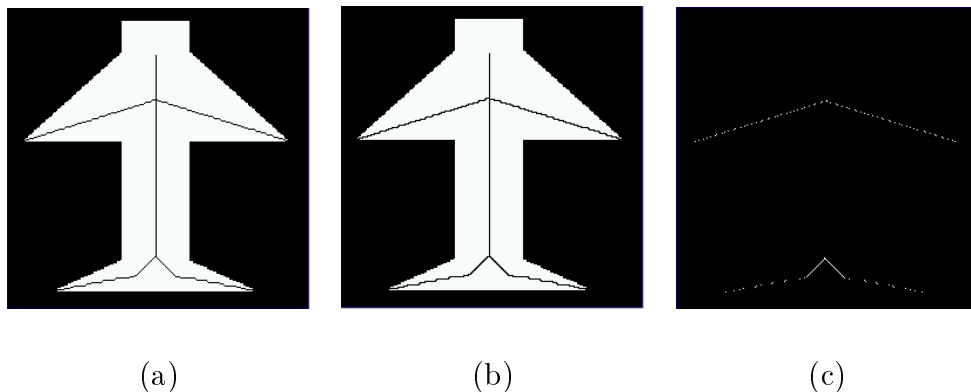


Figura 9.8: Esqueletos obtenidos por los algoritmos basados en (a) la conectividad de orden 8 y (b) de orden 4; (c) diferencias entre ambas versiones.

La figura 9.9 muestra los puntos del contorno de un pez que elimina el algoritmo basado en la conectividad de orden 8. No se incluye la imagen correspondiente a la versión de orden 4 porque no eliminó ningún punto de dicho contorno.

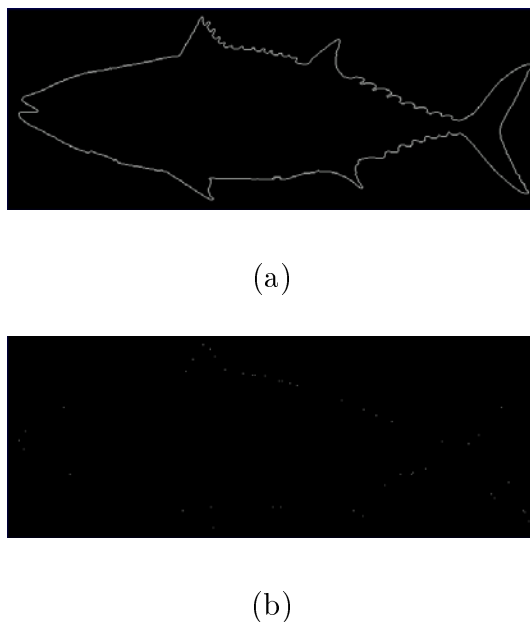


Figura 9.9: Adelgazamiento de un contorno: (a) contorno de una albacora; (b) puntos borrados por el algoritmo basado en la conectividad de orden 8.

Capítulo 10

Extracción del contorno exterior

10.1 Introducción

El algoritmo de comparación descrito en el capítulo 11 está basado en el análisis de los contornos exteriores de los objetos. Debido a ello, se ha diseñado una metodología que permite obtener el contorno más externo de un objeto contenido en una imagen (figura 10.1).

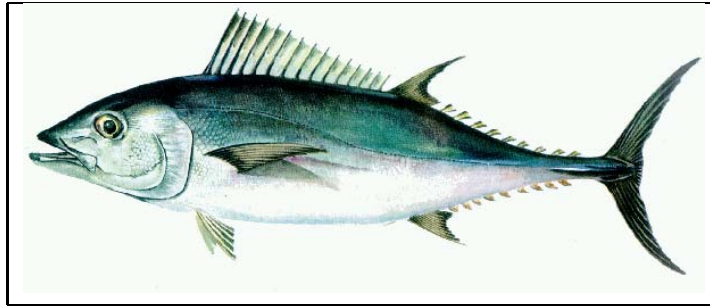
La metodología ha sido diseñada para obtener el contorno de imágenes que contienen un único objeto y consta de las siguientes fases:

1. Aplicación de un detector de bordes a la imagen.
2. Umbralización de la imagen.
3. Identificación de los puntos que pertenecen al contorno exterior del objeto contenido en la imagen.
4. Extracción del contorno exterior de la imagen.

Puesto que los algoritmos de detección de bordes y umbralización de imágenes ya son estudiados en profundidad en el capítulo 3, este capítulo sólo se va a ocupar de las otras dos fases.

La sección 10.2 describe las características y dificultades de los dos métodos que se han diseñado para identificar los puntos del contorno exterior de un objeto. Así mismo, en la sección 10.3 se explica el algoritmo que ha sido diseñado para extraer los contornos de los objetos contenidos en una imagen.

El objetivo final que se pretende con esta metodología es extraer de la imagen un único contorno exterior cerrado. Lamentablemente, existen situaciones en



(a)



(b)

Figura 10.1: Atún: (a) imagen en color y (b) contorno exterior.

las que la aplicación de los algoritmos anteriores provoca que el contorno exterior obtenido esté abierto, fragmentado o compuesto por puntos ruidosos. Para superar estas dificultades, se han diseñado los siguientes algoritmos auxiliares:

- Algoritmo para eliminar fragmentos aislados compuestos por puntos ruidosos (sección 10.3).
- Algoritmo para suprimir puntos que provocan “picos” en un contorno conexo (sección 10.3).
- Algoritmo para cerrar un contorno o unir sus fragmentos (sección 10.4).

En la sección 10.5 se detallan todas las fases necesarias para extraer de una imagen un único contorno cerrado.

10.2 Identificación de los puntos del contorno exterior

Una vez que se han detectado los bordes, se han de identificar los puntos que pertenecen al contorno exterior del objeto contenido en la imagen, para lo cual

se han desarrollado dos métodos denominados:

- Método del sondeo.
- Método de la pila.

A continuación se van a describir los dos métodos diseñados, explicándose su funcionamiento y dificultades.

10.2.1 Método del sondeo

El método del sondeo detecta los puntos del contorno exterior de una imagen mediante proyecciones o sondeos realizados desde los extremos superior, inferior, izquierdo y derecho de la imagen. Se considera que un punto pertenece al contorno exterior si no existe ningún punto marcado entre él y “alguno” de los límites de la imagen. La figura 10.2 ilustra el funcionamiento de este método.

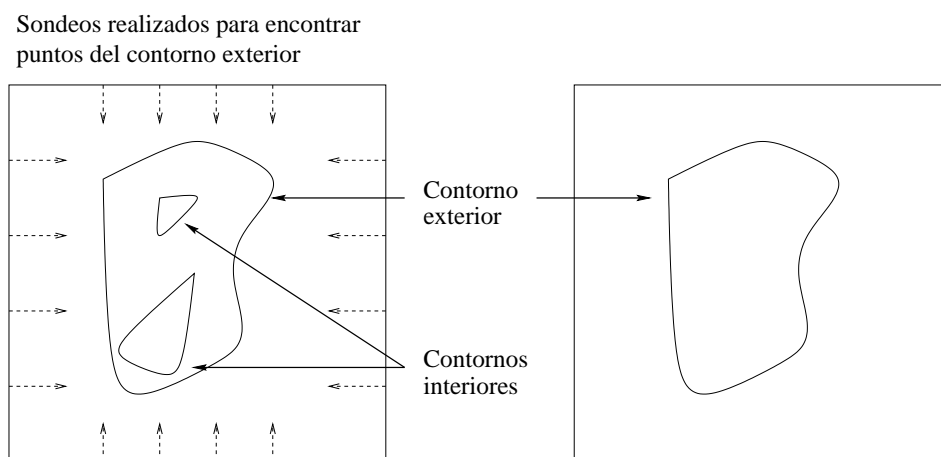


Figura 10.2: Identificación de los puntos del contorno exterior de un objeto mediante el método del sondeo.

Este método presenta algunos inconvenientes:

- Contorno exterior no cerrado: algunos puntos de los contornos interiores pueden ser considerados como puntos del contorno exterior.
- Ocultamientos: algunas zonas del contorno exterior puede no ser accesibles mediante el sondeo al estar ocultas por otras partes del contorno.
- Ruido: algunas zonas o puntos del contorno exterior pueden no ser accesibles por el sondeo debido a la presencia de puntos ruidosos fuera del contorno exterior.

Estos inconvenientes provocan que el contorno exterior sondeado no sea conexo y que presente puntos ruidosos. La figura 10.3 muestra cómo estas dificultades afectan al correcto funcionamiento del método del sondeo.

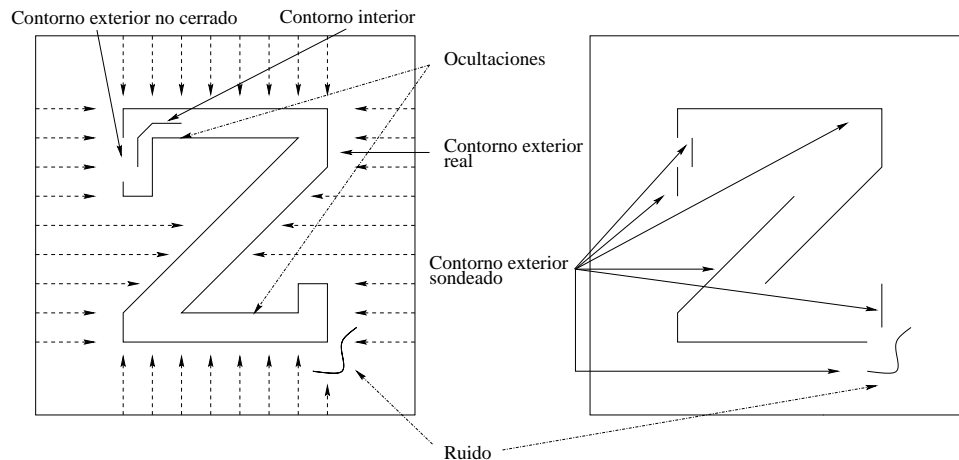


Figura 10.3: Dificultades del método del sondeo.

10.2.2 Método de la pila

Este método considera que un punto pertenece al contorno exterior de un objeto si existe un camino con “conectividad 4” entre dicho punto y alguno de los límites de la imagen (extremo superior, inferior, derecho o izquierdo). Se exige como restricción que el camino esté compuesto sólo por puntos que no pertenezcan al contorno (puntos no marcados), salvo el primero, que es el punto que pertenece al contorno exterior.

El nombre de este método se debe a que el algoritmo utiliza una pila para detectar los puntos del contorno exterior. A continuación se detalla el algoritmo desarrollado:

- [1] **Inicio**
- [2] Sea P el primer punto de la imagen que no esté marcado.
- [3] Marcar a P como punto visitado.
- [4] Introducir a P en la pila.
- [5] **Mientras** la pila no esté vacía **hacer**
- [6] Sea P el primer punto de la pila.
- [7] Quitar a P de la pila.
- [8] **Para** todo vecino (con conectividad 4)¹ no visitado Q de P **hacer**
- [9] **Si** Q está marcado
- [10] **entonces** marcar a Q como punto del contorno exterior
- [11] **si no** introducir a Q en la pila

¹Se utilizará un vecindario de 3×3 .

```

[12]         fin_si
[13]         Marcar a  $Q$  como punto visitado.
[14]         fin_para
[15]     fin_mientras
[16] fin

```

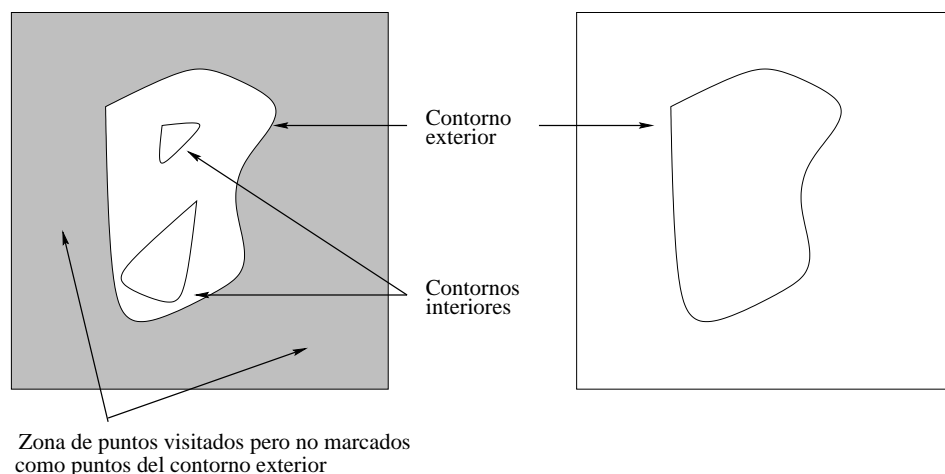


Figura 10.4: Identificación de los puntos del contorno exterior de un objeto mediante el método de la pila.

La figura 10.4 muestra los puntos que pertenecen al contorno exterior de un objeto contenido en una imagen.

El camino ha de respetar la conectividad 4 debido a que el uso de la conectividad 8 puede provocar “cruces” en el contorno, lo cual dificulta la posterior extracción de un único contorno (véase la sección 10.3).

La utilización de la conectividad 4 al aplicar el método de la pila permite obtener un contorno exterior con un número menor de puntos, que, además, están conectados entre sí mediante la conectividad 8. La figura 10.5 muestra cómo influye el uso de la conectividad 8 ó 4 en la identificación de los puntos del contorno exterior de un objeto.

No obstante, el uso de la conectividad 4 no evita que este método de identificación de los puntos del contorno exterior presente dos inconvenientes o dificultades (figura 10.6):

- Contorno exterior no cerrado: esta es la principal dificultad de este método ya que provoca que contornos interiores sean considerados como contornos exteriores.
- Ruido: los puntos situados fuera del contorno exterior que estén marcados también son considerados como puntos del contorno exterior.

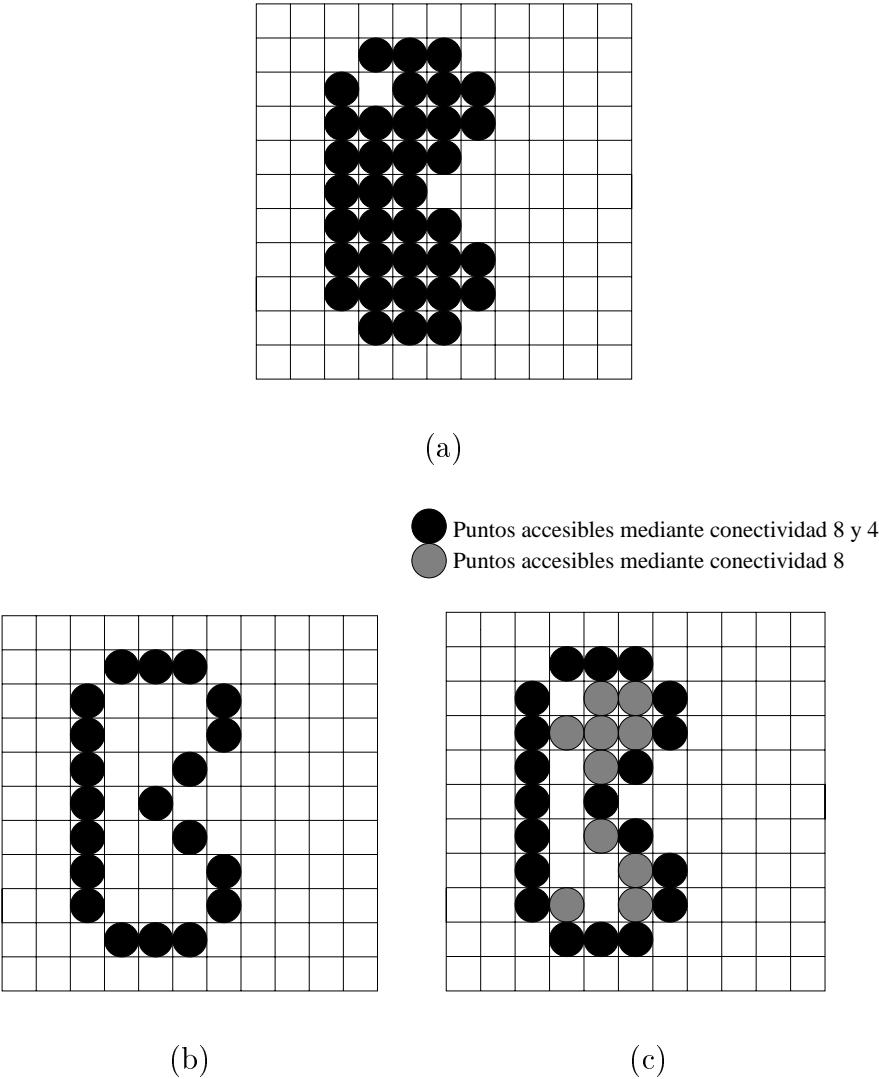


Figura 10.5: Justificación del uso de la conectividad 4 al identificar los puntos del contorno exterior con el método de la pila: (a) imagen original; puntos obtenidos utilizando (b) conectividad 4 y (c) conectividad 8.

10.2.3 Comparación entre los métodos

Si se compara el rendimiento de ambos métodos, se pueden distinguir dos casos:

1. El contorno exterior de la imagen está abierto: el método del sondeo supera al método de la pila, ya que éste considera como puntos del contorno exterior a puntos interiores (figuras 10.6 y 10.7(c)). Sin embargo, el método del sondeo tiene problemas debido a los ocultamientos (figuras 10.3 y 10.7(b)).
2. El contorno exterior de la imagen está cerrado: el método de la pila identifica a todos los puntos del contorno (figura 10.8(c)), mientras que el método del sondeo tiene problemas debido a los ocultamientos, lo cual provoca que

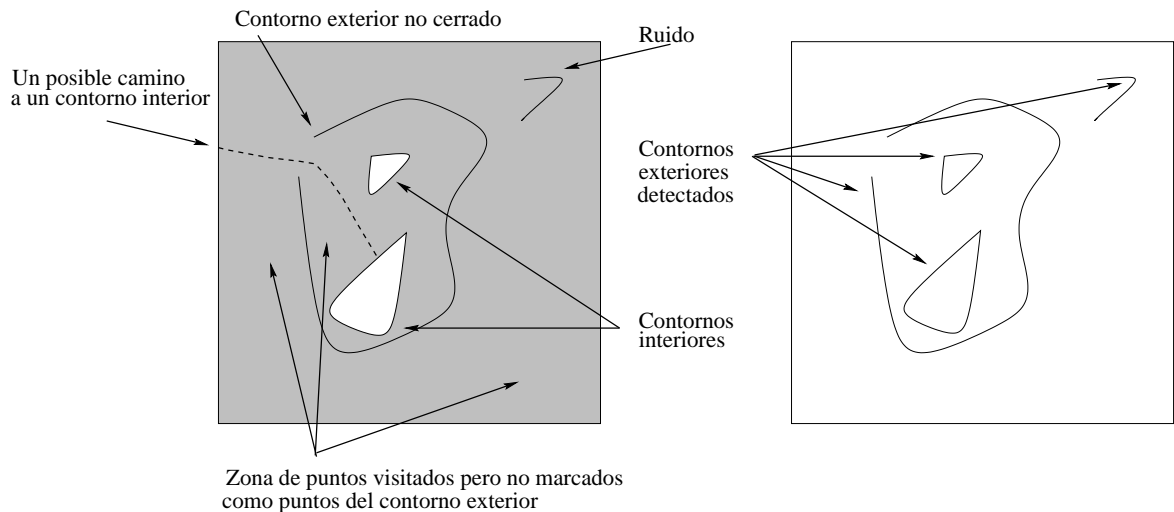


Figura 10.6: Dificultades del método de la pila.

el contorno obtenido esté fragmentado (figura 10.8(b)).

Como se puede observar, en ambas situaciones el método del sondeo tiene problemas debido a los ocultamientos, lo cual induce a pensar que el método de la pila es mejor. Sin embargo, este último método tiene un rendimiento muy deficiente cuando el contorno exterior de la imagen no está cerrado.

En resumen, es importante destacar que si el detector de bordes es capaz de generar un contorno exterior cerrado, entonces se ha de aplicar el método de la pila; en caso contrario, el método que se ha de utilizar es el método del sondeo.

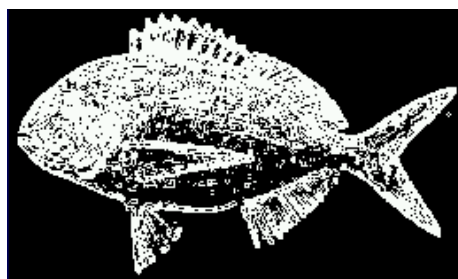
10.3 Extracción de los contornos de una imagen

El algoritmo de extracción de contornos trata de obtener todos los contornos de la imagen, ordenando sus puntos para que puedan ser procesados posteriormente.

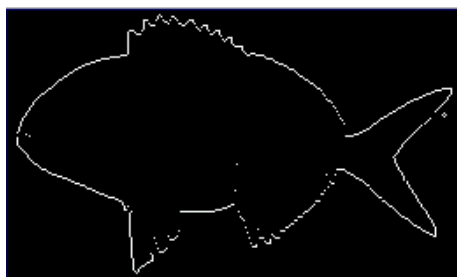
El algoritmo comienza seleccionando el punto marcado de la imagen que tenga menos vecinos marcados. El punto elegido se inserta como primer punto de un nuevo contorno. Este punto también es considerado como el punto final “actual”². A continuación, el algoritmo incorpora al contorno a aquel punto de la imagen que cumpla las siguientes condiciones:

- El punto ha de estar marcado,
- sea vecino del punto final “actual” del contorno

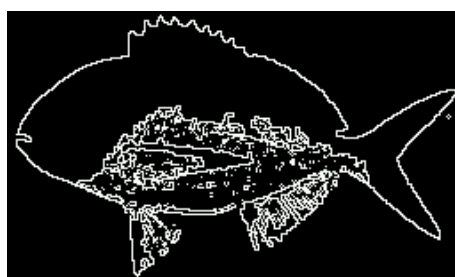
²Se va a utilizar el término “actual” para reflejar la situación de la imagen o el contorno en cada paso de ejecución del algoritmo.



(a)

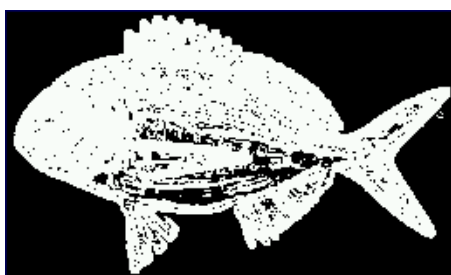


(b)



(c)

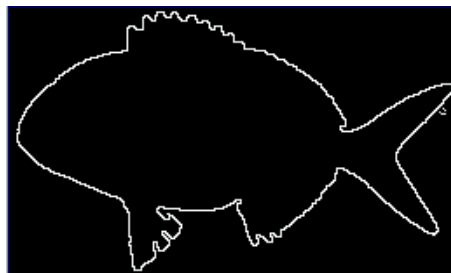
Figura 10.7: Funcionamiento de los métodos de identificación de los puntos del contorno exterior cuando éste está abierto: (a) imagen original; puntos del contorno exterior identificados por los métodos del sondeo (b) y de la pila (c).



(a)



(b)



(c)

Figura 10.8: Funcionamiento de los métodos de identificación de los puntos del contorno exterior cuando éste está cerrado: (a) imagen original; puntos del contorno exterior identificados por los métodos del sondeo (b) y de la pila (c).

- y tenga el menor número de vecinos “actuales” marcados.

Si hay varios puntos que cumplan estos requisitos, se elige a aquel punto que haya sido localizado primero según el siguiente orden: norte, noreste, este, sureste, sur, suoreste, oeste y noroeste.

El punto seleccionado se borra de la imagen y pasa a ser el punto final “actual” del contorno. La inserción de nuevos puntos en un contorno termina cuando el punto final “actual” no tenga ningún vecino marcado.

Si quedan puntos marcados en la imagen, se crea un nuevo contorno y se repite el procedimiento ya descrito. El algoritmo concluye cuando no quede ningún punto marcado en la imagen.

A continuación se detalla el algoritmo utilizado para la extracción de todos los contornos de la imagen:

```

[1] Inicio
[2]   Mientras haya algún punto marcado en la imagen hacer
[3]     Crear un nuevo contorno que pasa a ser el contorno actual.
[4]     Hallar el primer punto marcado P con menos vecinos marcados.
[5]     Borrar el punto P de la imagen.
[6]     Insertar a P al final del contorno actual.
[7]     Convertir a P en el punto final “actual”.
[8]     Mientras que el punto final “actual” tenga algún vecino marcado
[9]       hacer
[10]        Hallar el vecino marcado Q del punto final “actual”
[11]        con menos vecinos marcados3.
[12]        Borrar el punto Q de la imagen.
[13]        Insertar el punto Q al final del contorno actual.
[14]        Convertir a Q en el punto final “actual”.
[15]      fin_mientras
[16]      Guardar el contorno actual.
[17]    fin_mientras
[18]  fin

```

La figura 10.9 muestra algunos ejemplos del funcionamiento de este algoritmo de extracción de contornos.

Como se puede observar en las figuras 10.9(a) y (b), el sentido de recorrido de los contornos cerrados depende de la conectividad de los puntos. Para evitar este problema, se necesita aplicar antes una fase de adelgazamiento de bordes con conectividad 8, consiguiéndose así que el recorrido de los contornos cerrados sea

³Si hay varios puntos con el mismo número de vecinos marcados, se elige el primero que se haya localizado según el siguiente orden: norte, noreste, este, sureste, sur, suoreste, oeste y noroeste.

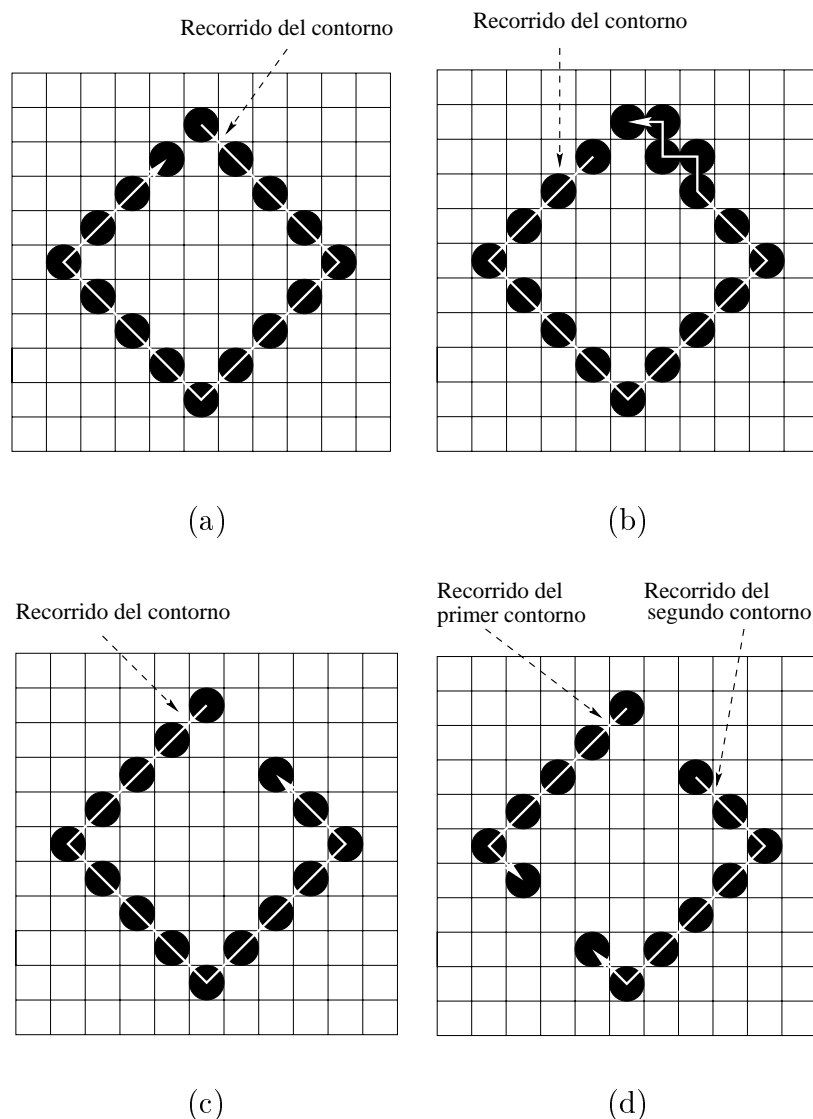
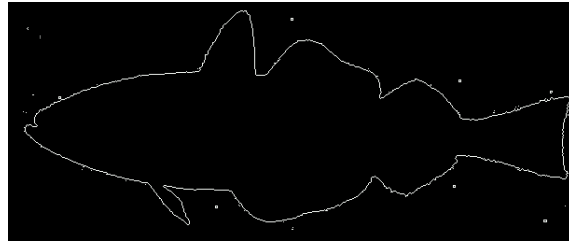


Figura 10.9: Recorridos de los contornos obtenidos por el método de extracción: (a) y (b) contornos cerrados; (c) contorno abierto; (d) dos contornos.

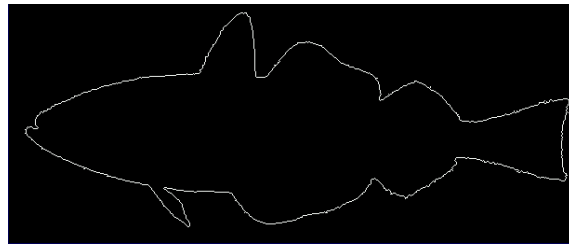
siempre en el sentido horario.

Se pretende utilizar este algoritmo para extraer un único contorno de la imagen. Obviamente, el algoritmo extraerá más de un contorno si el contorno de la imagen no es conexo (figura 10.9(d)). Este problema es provocado por las dificultades de los algoritmos de identificación de los puntos del contorno exterior, como ya se indicó en la sección 10.2. Para resolver este problema, se ha diseñado un algoritmo auxiliar, denominado “eliminar_contornos”, que elimina aquellos contornos cuyo número de puntos sea inferior a una cota mínima preestablecida. La figura 10.10 muestra la aplicación de este algoritmo.

También hay otras situaciones en las que, aun siendo el contorno cerrado, se puede extraer más de un contorno. Ello es debido a la presencia en la imagen



(a)



(b)

Figura 10.10: Eliminación de contornos con pocos puntos: (a) imagen con contornos con puntos ruidosos; (b) imagen obtenida al eliminar los contornos con menos de 20 puntos.

de “picos” o “cruces” como los que aparecen en la figura 10.11.

Para evitar los problemas planteados por la presencia de “picos” en contornos cerrados, se ha diseñado un algoritmo auxiliar, denominado “suprimir_picos”, que permite suprimir dichos “picos” mediante un proceso iterativo que elimina los puntos que, a lo sumo, tengan un vecino “actual” marcado. El proceso de supresión concluye cuando no se elimina ningún punto en una iteración (figura 10.12).

El funcionamiento del algoritmo de supresión de picos es correcto si el extremo de cada pico tiene, a lo sumo, un único vecino marcado. Sin embargo, hay situaciones en las que esto no ocurre, como se puede comprobar en la figura 10.13(a) donde los extremos de los picos tienen dos vecinos marcados, y, por tanto, no serían eliminados por el algoritmo. Para solucionar este problema, se ha de aplicar previamente un algoritmo de adelgazamiento de bordes que utilice la conectividad 8, como los descritos en el capítulo 5 (figura 10.13).

Por último, se ha de indicar que la supresión de los “cruces” es mucho más compleja y requiere el uso de más información heurística.

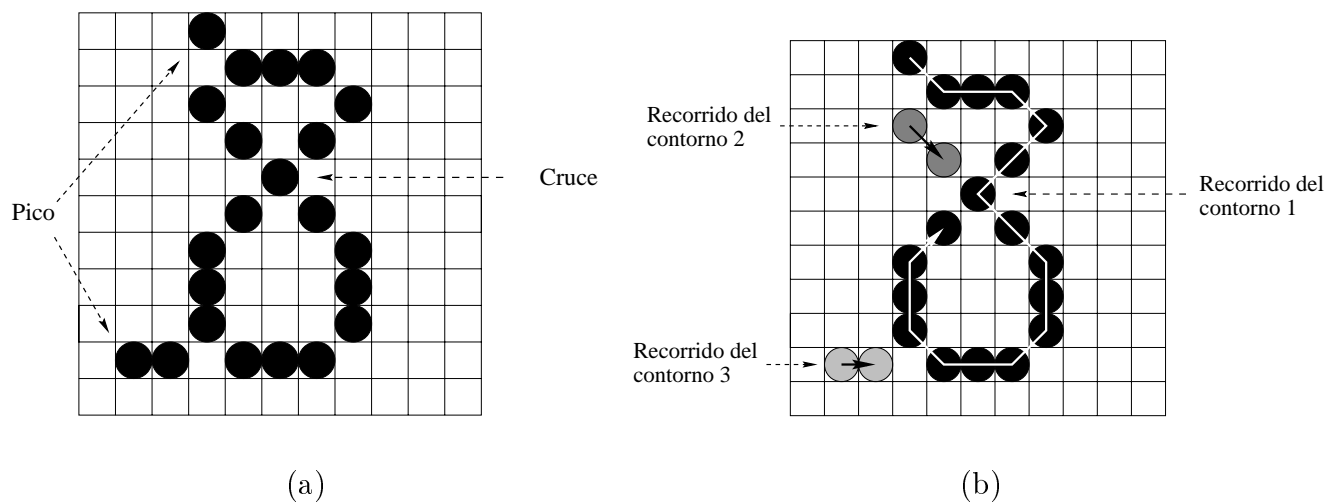


Figura 10.11: Dificultades del método de extracción: (a) contorno con “picos” y “cruce”; (b) contornos extraídos.

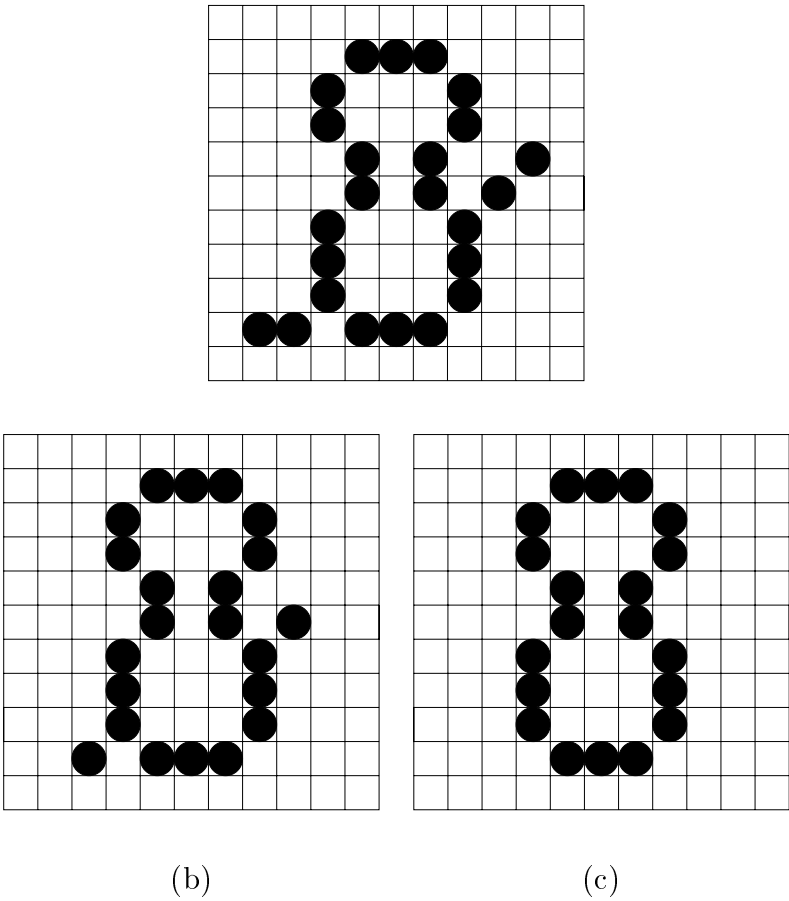
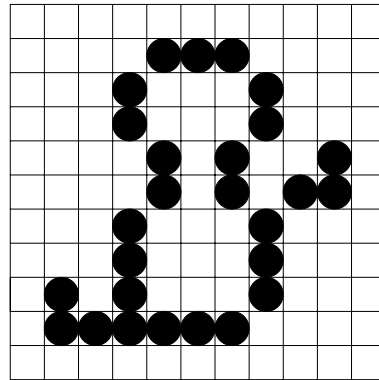
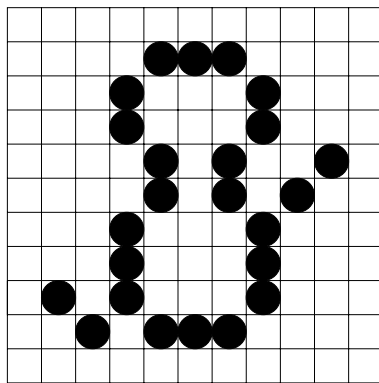


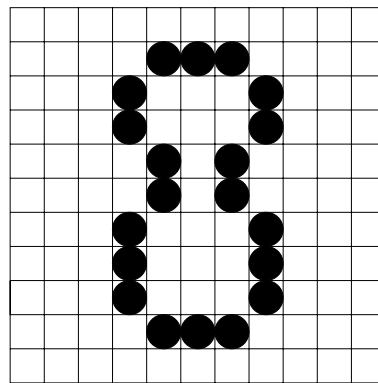
Figura 10.12: Aplicación del algoritmo de supresión de “picos”: (a) imagen original; (b) primera iteración y (c) segunda iteración.



(a)



(b)



(c)

Figura 10.13: Mejora del algoritmo de supresión de “picos”: (a) imagen original; (b) adelgazamiento de (a) con un algoritmo con conectividad 8 y (c) supresión de “picos” de (b).

10.4 Unión de los contornos de una imagen

El algoritmo de comparación de objetos descrito en el capítulo 11 necesita que se haya extraído previamente un único contorno de cada una de las imágenes. Dicho contorno puede ser abierto o cerrado, aunque el algoritmo de comparación sólo se ha aplicado a contornos cerrados. Debido a ello, si una imagen contiene un contorno abierto o fragmentado en varios contornos entonces se ha de usar un algoritmo de unión de contornos que genere un único contorno cerrado.

La figura 10.14 muestra dos imágenes con contornos cerrados que no requieren la aplicación del algoritmo de unión de contornos.

El algoritmo de unión de contornos diseñado ha tenido en cuenta las siguientes restricciones:

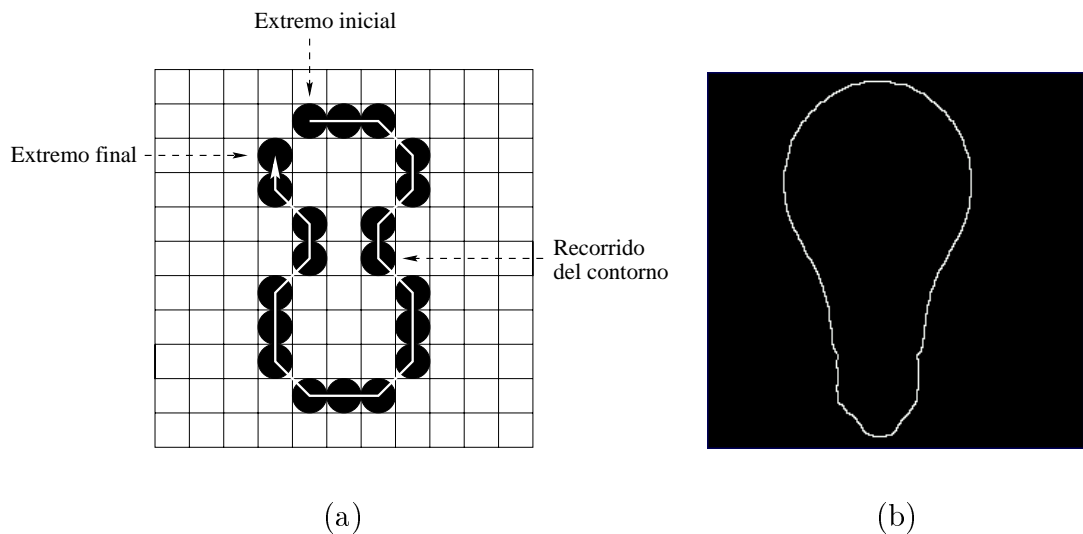


Figura 10.14: Ejemplos de contornos cerrados: (a) puntos extremos y recorrido de un contorno cerrado; (b) contorno exterior cerrado de una bombilla.

- Todos los contornos de una imagen han de formar parte de un único contorno cerrado que los contenga.
- El contorno cerrado que se genere no debe tener cruces.

La segunda restricción es necesaria para evitar que el algoritmo de extracción de contornos obtenga más de un contorno de la imagen (figura 10.11).

El algoritmo diseñado es el siguiente:

```

[1] Inicio
[2]   Extraer los contornos de la imagen (sección 10.3).
[3]   Según el número de contornos hacer
[4]   0: Escribir("No hay contornos en la imagen")
[5]   1:
[6]     Si el contorno no está cerrado
[7]       entonces unir el extremo inicial y el extremo final
[8]     fin_si
[9]   otro:
[10]    Mientras haya más de un contorno hacer
[11]      Determinar los dos puntos extremos de dos contornos
[12]      diferentes que disten menos entre sí.
[13]      Unir los puntos extremos obtenidos.
[14]      Convertir en uno los dos contornos unidos.
[15]    fin_mientras
[16]    Unir los extremos del contorno resultante
[17]    Devolver el contorno
[18]  fin_según
[19] fin

```

Es muy importante destacar que, cuando hay más de un contorno, la sentencia de las líneas 11 y 12 requiere que los puntos a unir pertenezcan a contornos diferentes, evitando así que se pueda generar más de un contorno cerrado.

Para efectuar las uniones de los puntos extremos que se indican en las líneas 7, 13 y 16, se han codificado dos algoritmos (Hearn y Baker 1994):

- Unión con un segmento recto: se ha utilizado el algoritmo de línea de Bresenham.
- Unión con “b-splines” cúbicos.

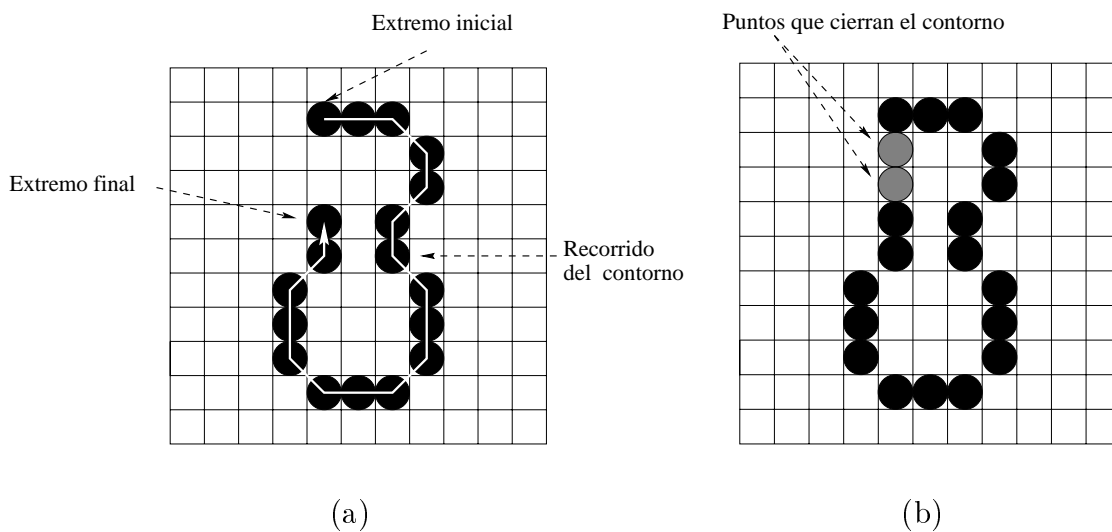


Figura 10.15: Cierre de un contorno abierto: (a) recorrido de un contorno abierto; (b) cierre del contorno (a).

Las figuras 10.15 y 10.16 muestran el cierre de contornos abiertos. Las figuras 10.17 y 10.18 muestran los pasos desarrollados por el algoritmo de unión de contornos cuando hay más de un contorno en la imagen.

La aplicación del algoritmo de unión de contornos puede provocar que se cree un contorno cerrado pero con “picos”, lo cual dificultaría el cometido de extraer un único contorno de la imagen. Como ya se indicó en la sección 10.3, este problema se resuelve aplicando un algoritmo de adelgazamiento de bordes con conectividad 8 y, a continuación, el algoritmo de supresión de picos (figura 10.19).

Lamentablemente, en ciertas ocasiones, el algoritmo de unión de contornos puede generar algún “cruce” como los que se muestran en las figuras 10.20 y 10.21. En la sección 10.3 ya se comentó cómo los cruces impiden extraer un único contorno cerrado de la imagen, que es el objetivo final que se pretende conseguir con esta metodología. La solución de este problema es mucho más compleja y

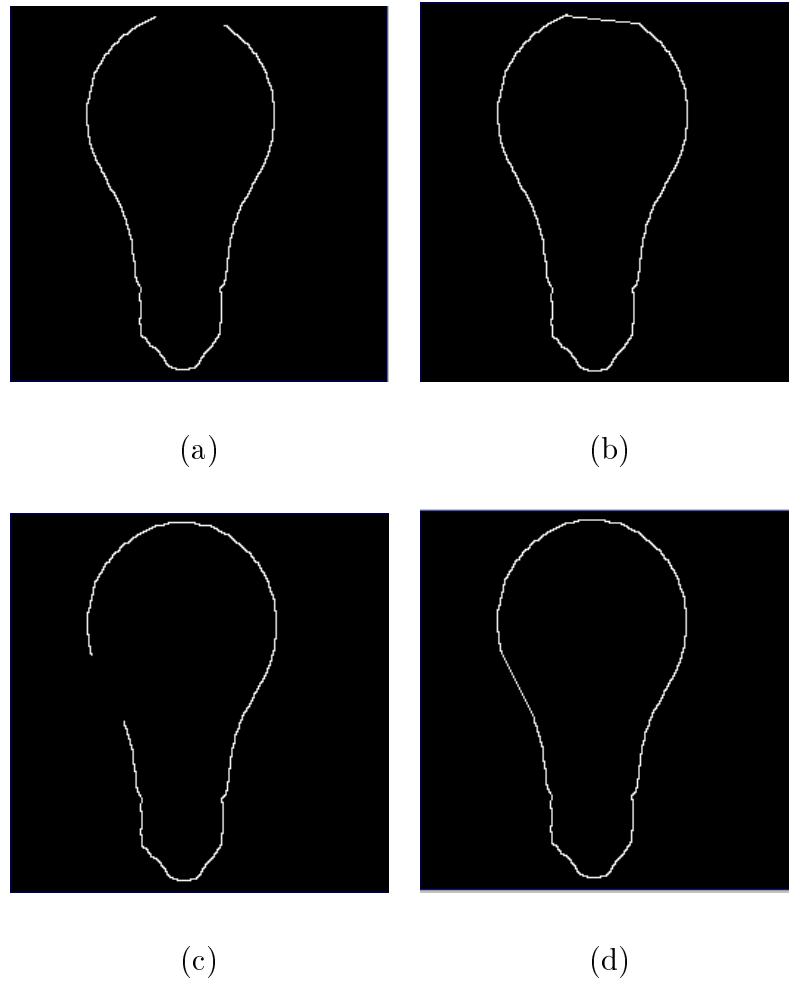


Figura 10.16: Cerramiento de contornos abiertos de una bombilla: (a) contorno con una apertura superior; (b) cierre del contorno (a); (c) contorno con una apertura lateral; (d) cierre del contorno (c).

suele aparecer si se utiliza el método del sondeo para identificar los puntos del contorno exterior. Debido a ello, se ha de procurar utilizar, en cambio, el método de la pila, ya que reduce, aunque no elimina completamente, la posibilidad de la aparición de cruces en el contorno. Aquí se manifiesta con claridad la influencia que ejercen las fases de detección de bordes y umbralización, ya que son ellas las que permitirán o no la aplicación del método de la pila (véase la subsección 10.2.3).

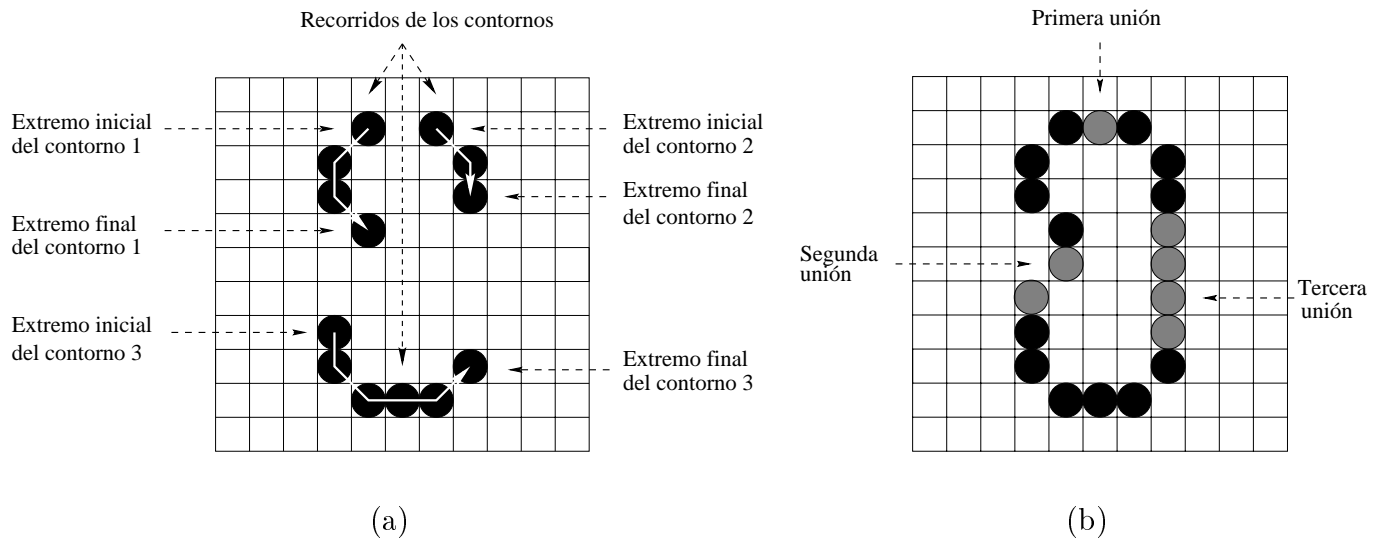


Figura 10.17: Pasos de la unión de tres contornos.

10.5 Resumen de la metodología de extracción del contorno exterior

Después de haber analizado detenidamente el problema de extracción del contorno exterior de un objeto contenido en una imagen, a continuación se reseña la metodología que ha sido diseñada:

1. Aplicación de un detector de bordes a la imagen.
2. Umbralización de la imagen.
3. Identificación de los puntos del contorno exterior:
 - Método del sondeo.
 - Método de la pila.
4. Adelgazamiento con conectividad 8 del contorno.
5. Extracción de los contornos de la imagen.
6. Eliminación de los contornos con pocos puntos.
7. Unión de los contornos.
8. Adelgazamiento con conectividad 8 del contorno cerrado.
9. Supresión de picos en el contorno cerrado.
10. Extracción del contorno final.

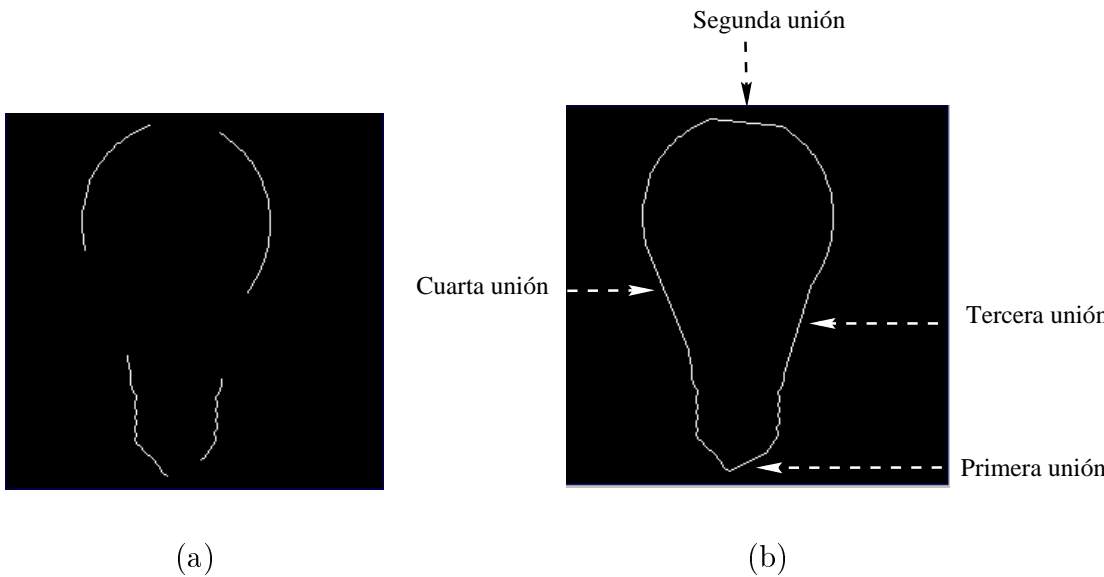


Figura 10.18: Unión de varios contornos: (a) contornos de una bombilla; (b) cierre del contorno (a).

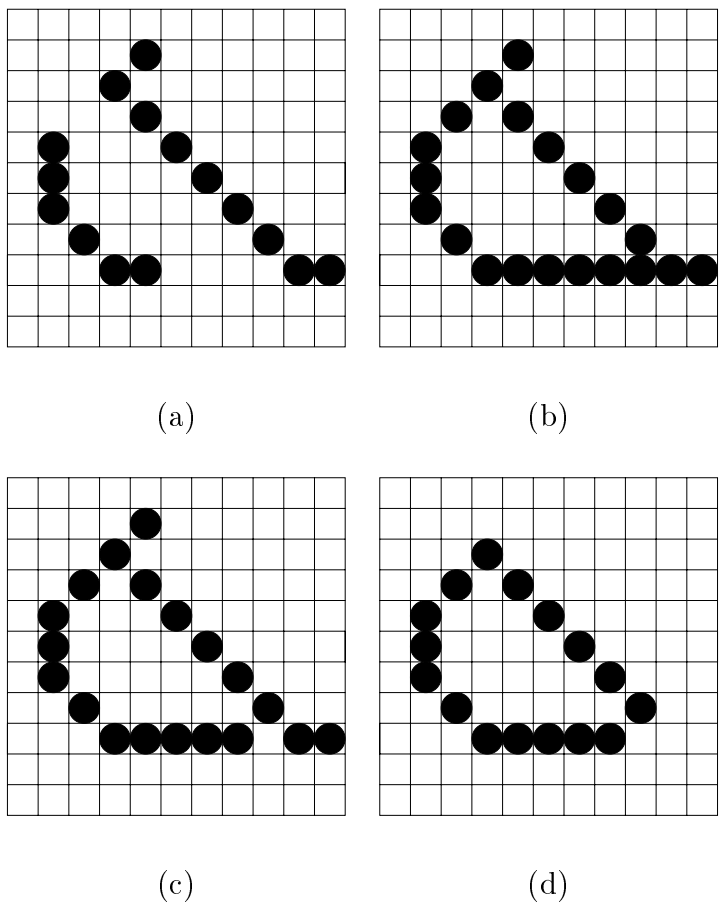


Figura 10.19: Mejora del método de unión de varios contornos: (a) contornos de un objeto; (b) unión de los contornos de (a); (b) adelgazamiento con conectividad 8 de (b); (c) supresión de “picos” de (c).

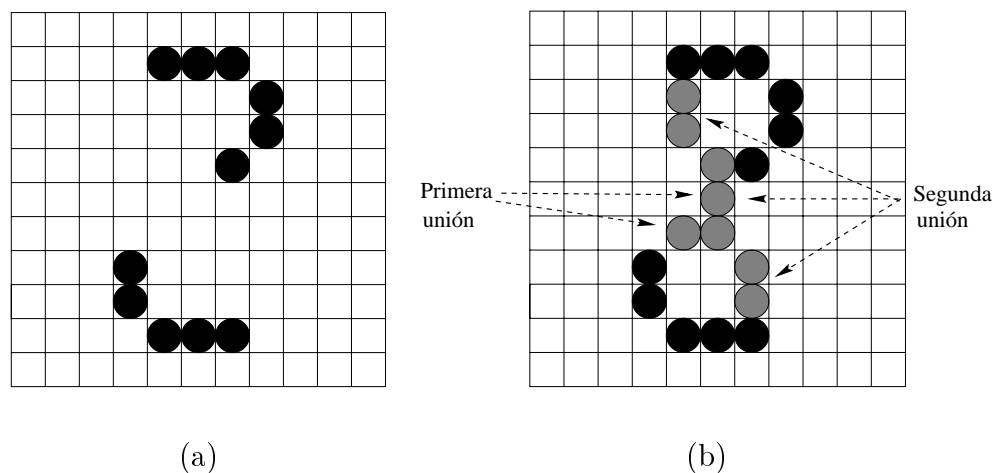


Figura 10.20: Dificultades en la unión de contornos: (a) contornos de una imagen; (b) cruce provocado por el algoritmo de unión.

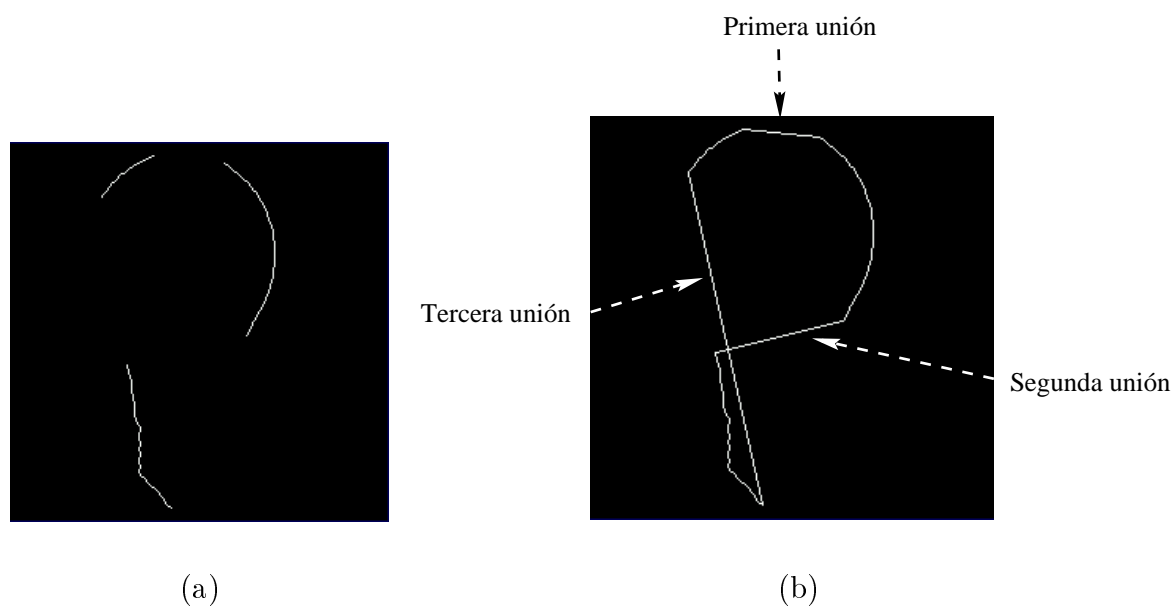


Figura 10.21: Dificultades del algoritmo de unión de contornos: (a) imagen con varios contornos; (b) unión de los contornos de la imagen (a) que provoca un cruce.

Capítulo 11

Algoritmo de comparación de contornos

11.1 Introducción

Se ha desarrollado un algoritmo que permite comparar dos objetos entre sí mediante el uso de los puntos críticos, característicos o relevantes de sus contornos.

La aplicación de este algoritmo de comparación requiere que se hayan extraído previamente los contornos exteriores de cada uno de los objetos, para lo cual se debe haber utilizado la metodología de extracción del contorno exterior desarrollada en el capítulo 10.

Para poder procesar los contornos, se necesita que sus puntos estén parametrizados. La sección 11.2 explica el método desarrollado.

El uso de los puntos críticos para caracterizar un contorno ha sido propuesto en numerosas ocasiones como ya se ha indicado en el capítulo 6. Los puntos críticos utilizados en el presente algoritmo de comparación se obtienen a partir de los puntos de curvatura cero, máxima, mínima o extrema (máxima o mínima) de los contornos. La subsección 11.3.2 explica cómo se obtienen tales puntos. Para estimar la curvatura en un punto se ha utilizado el algoritmo de Mokhtarian y Mackworth (1992), el cual es descrito en la subsección 11.3.1.

Una vez obtenidos los puntos críticos de cada contorno, el algoritmo de comparación trata de minimizar una función basada en el algoritmo de mínimos cuadrados propuesto por Gu y Tjahjadi (2.000). La sección 11.4 describe dicha función y el método de minimización.

Por último, la sección 11.5 indica los inconvenientes del algoritmo de Gu y Tjahjadi y se propone una mejora de dicho algoritmo.

11.2 Parametrización de los contornos

Para poder procesar un contorno, se necesita que éste esté parametrizado, es decir, sus puntos han de estar numerados a partir de un punto inicial y teniendo en cuenta un sentido de recorrido.

Un método básico de parametrización de un contorno consiste en seleccionar como primer punto del contorno a aquél que se localice en primer lugar al recorrer la imagen de arriba a abajo y de izquierda a derecha, y, a continuación, se numeran los demás puntos siguiendo el sentido de las agujas del reloj.

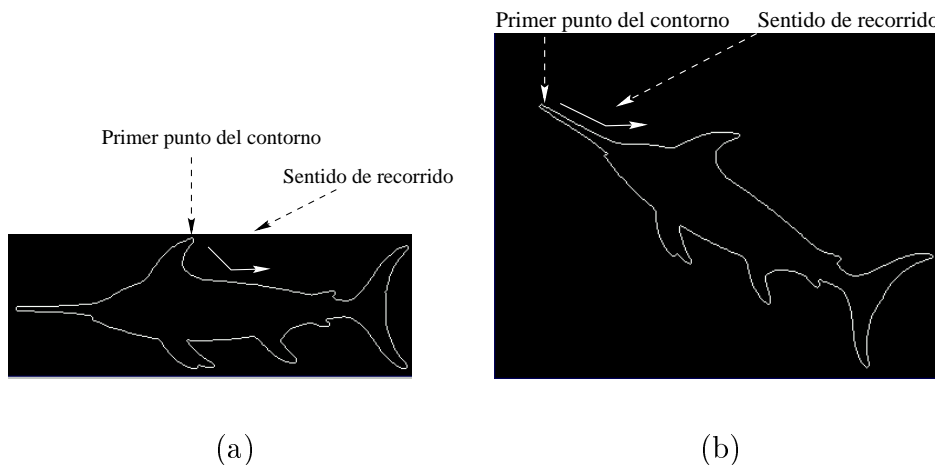


Figura 11.1: Dificultades del método básico de parametrización de un contorno: (a) recorrido del contorno original; (b) recorrido del contorno original rotado 30° .

Este método presenta un gran inconveniente: no es invariante a rotaciones. La figura 11.1 muestra cómo influye la rotación en la determinación del primer punto del contorno.

Para superar la dificultad que conlleva determinar el primer punto del contorno, se propone el siguiente método:

1. Calcular el centroide del contorno.
2. Calcular el eje de mínima inercia del contorno (figura 11.2).
3. Localizar los puntos del contorno que pertenezcan al eje de mínima inercia y considerar como primer punto del contorno a aquél que diste más del centroide.

La digitalización de la imagen puede provocar que la intersección del contorno con el eje de inercia sea nula (figura 11.3). Debido a ello, se modifica el criterio de selección del primer punto del contorno de la siguiente forma:

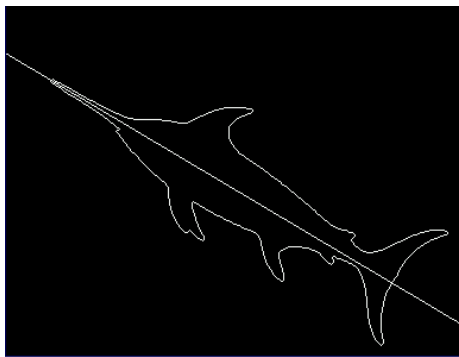


Figura 11.2: Eje de mínima inercia.

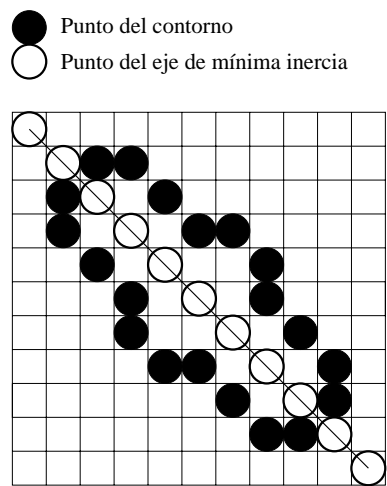


Figura 11.3: Eje de mínima inercia que no intersecta al contorno.

1. Obtener los puntos del contorno que disten menos del eje de mínima inercia (figura 11.4).
2. Entre los puntos obtenidos en el paso 1, considerar como primer punto a aquél cuya distancia respecto del centroide sea mayor.

Obviamente, esta modificación tiene en cuenta el caso en que la intersección entre el contorno y el eje de mínima inercia no sea nula, ya que entonces la distancia mínima al eje será cero.

No obstante lo anterior, existen situaciones en las que este método de selección del primer punto presenta la siguiente dificultad: el centroide puede equidistar de las zonas en las que se produce la intersección entre el contorno y el eje de mínima inercia, provocando que algunas veces se considere como primer punto a uno que pertenezca a una zona y otras veces se considere como primer punto a otro que esté situado en una zona diametralmente opuesta.

Para resolver este problema, originado por la forma geométrica del contorno,

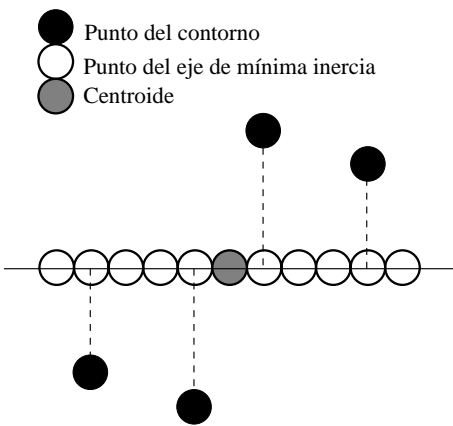
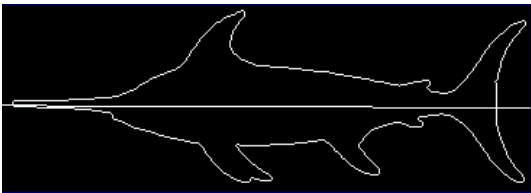


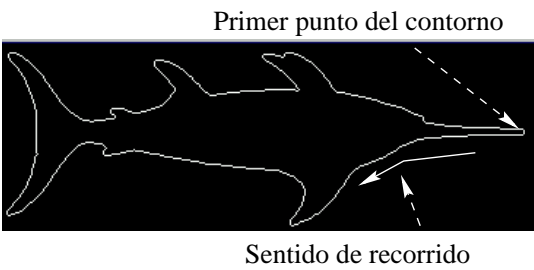
Figura 11.4: Distancia de los puntos del contorno al eje de mínima inercia.

se ha optado por utilizar los dos posibles puntos candidatos como puntos iniciales, generándose así dos parametrizaciones diferentes del mismo contorno.

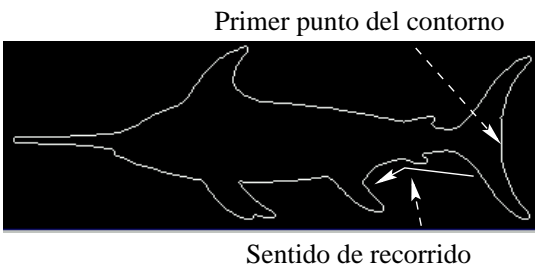
Por tanto, será necesario tener en cuenta las dos parametrizaciones de cada uno de los contornos para poder compararlos entre sí.



(a)



(b)



(c)

Figura 11.5: Parametrizaciones de un contorno: (a) eje de mínima inercia; (b) primera parametrización; (c) segunda parametrización.

Además, cada contorno va a ser rotado para que el primer punto de la parametrización tenga un ángulo (en coordenadas polares respecto del centroide) igual a cero.

La figura 11.5 muestra las dos parametrizaciones de un contorno y sus recorridos según el sentido de las agujas del reloj.

Hay que destacar que el método de parametrización que se propone es invariante a rotaciones y traslaciones, aunque conlleve un mayor coste computacional.

11.3 Puntos críticos o relevantes de un contorno

11.3.1 Estimación de la curvatura en un punto: algoritmo de Mokhtarian y Mackworth

El algoritmo de comparación desarrollado utiliza los puntos de curvatura cero, máxima, mínima o extrema (máxima o mínima) de los contornos. Para detectar tales puntos, se ha aplicado previamente el algoritmo propuesto por Mokhtarian y Mackworth (Mokhtarian y Mackworth 1992; Mokhtarian 1995) para estimar la curvatura en un punto.

El funcionamiento del algoritmo de Mokhtarian y Mackworth es el siguiente:

Considérese que el contorno ha sido parametrizado como una curva Γ :

$$\Gamma(u) = (x(u), y(u)) \quad (11.1)$$

y que a partir de Γ se obtiene una curva suavizada Γ_σ definida como

$$\Gamma_\sigma(u) = (X_\sigma(u), Y_\sigma(u)) \quad (11.2)$$

donde

$$X_\sigma(u) = x(u) \star G_\sigma(u) \quad (11.3)$$

y

$$Y_\sigma(u) = y(u) \star G_\sigma(u) \quad (11.4)$$

donde “ \star ” denota el operador de convolución y $G_\sigma(u)$ es la función gaussiana de anchura σ (Marr y Hildreth 1980):

$$G_\sigma(u) = \frac{1}{\sqrt{2\pi} \sigma} e^{\frac{-u^2}{2\sigma^2}} \quad (11.5)$$

Se puede estimar la curvatura en un punto como:

$$\kappa_\sigma(u) = \frac{X'_\sigma(u)Y''_\sigma(u) - Y'_\sigma(u)X''_\sigma(u)}{(X'_\sigma(u)^2 + Y'_\sigma(u)^2)^{1.5}} \quad (11.6)$$

donde se han utilizado las siguientes aproximaciones

$$X'_\sigma(u) = X_\sigma(u) \star G'_\sigma(u) \quad (11.7)$$

$$Y'_\sigma(u) = Y_\sigma(u) \star G'_\sigma(u) \quad (11.8)$$

y

$$X''_\sigma(u) = X_\sigma(u) \star G''_\sigma(u) \quad (11.9)$$

$$Y''_\sigma(u) = Y_\sigma(u) \star G''_\sigma(u) \quad (11.10)$$

siendo G'_σ y G''_σ la primera y segunda derivada, respectivamente, de la función gaussiana de anchura σ .

El uso de la función gaussiana $G_\sigma(u)$ permite crear “un espacio de escala de curvatura” de forma que se puede efectuar un análisis multiescala de los contornos.

La figura 11.6 muestra los puntos de curvatura cero y extrema de un contorno obtenidos mediante la aplicación del algoritmo de Mokhtarian y Mackworth con $\sigma = 5$.

Como se puede observar en las figuras 11.6(d) y (f), la aproximación lineal del contorno original que se obtiene a partir de los puntos de curvatura extrema (máximos o mínimos) es superior a la obtenida a partir de los puntos de curvatura cero.

11.3.2 Puntos críticos utilizados en el algoritmo de comparación

El algoritmo de comparación está basado en los puntos críticos (característicos o relevantes) de cada uno de los dos contornos que se van a comparar. En este algoritmo se ha considerado que un punto de un contorno es crítico si cumple alguna de las dos siguientes condiciones:

- Es un punto de curvatura cero, máxima, mínima o extrema (máxima o mínima)¹.
- Es un punto “equivalente” a un punto de curvatura cero, máxima, mínima o extrema (máxima o mínima) del otro contorno.

Dos puntos de dos contornos diferentes son “equivalentes” si ocupan la misma posición relativa desde el principio de sus respectivos contornos. Considérese, por ejemplo, que los puntos de los contornos se numeran desde 0 hasta $N - 1$, donde N es el número total de puntos del contorno. Si Q es un punto del segundo contorno que ocupa la posición i respecto del principio de su contorno, entonces el punto P del primer contorno que es equivalente a Q ha de ocupar la posición

$$j = \frac{i \times (N_1 - 1)}{N_2 - 1} \quad (11.11)$$

¹Al aplicar el algoritmo, se deberá indicar qué tipo de puntos de curvatura se desea utilizar.

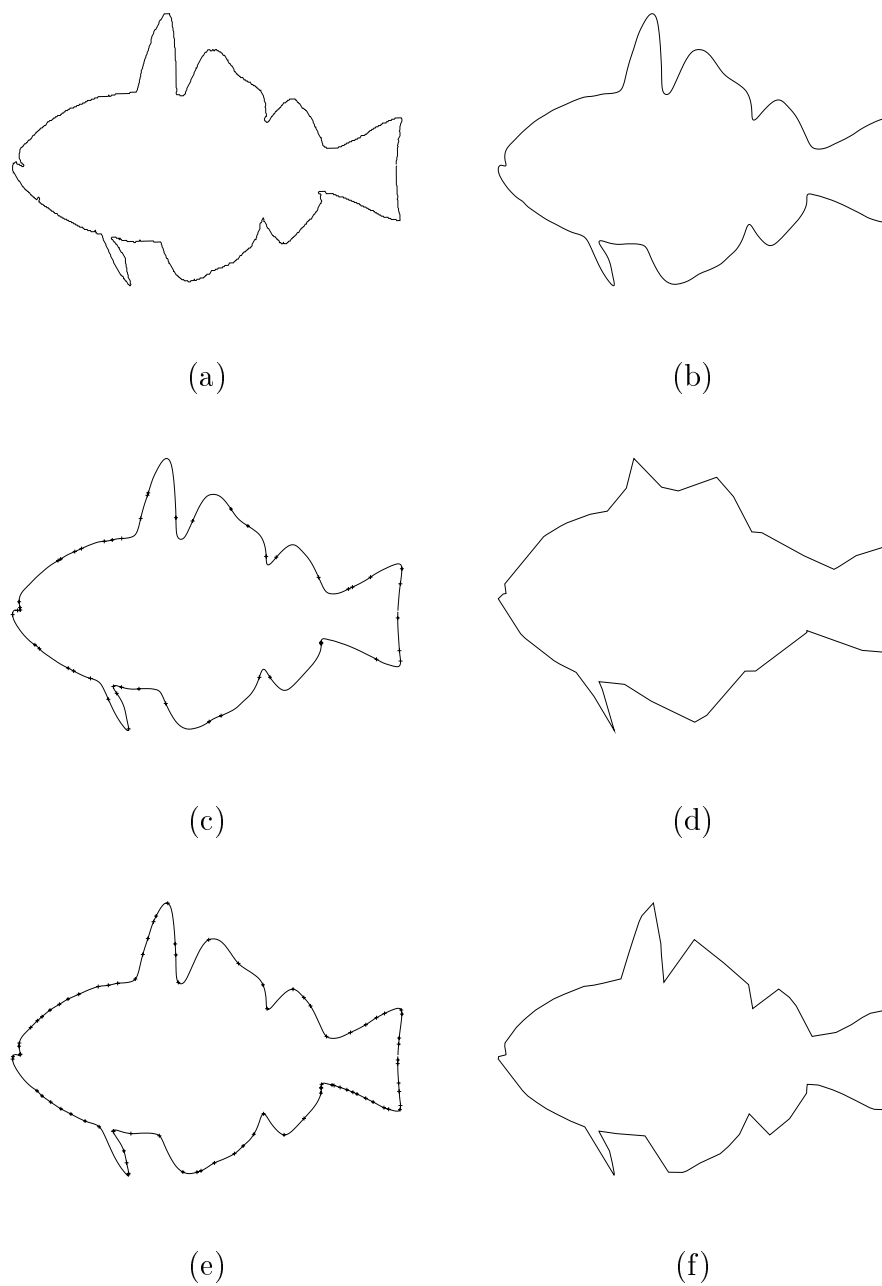


Figura 11.6: Aplicación del algoritmo de Mokhtarian y Mackworth: (a) contorno original; (b) contorno suavizado con el filtro gaussiano ($\sigma = 5$); (c) puntos de curvatura cero sobre el contorno suavizado; (d) aproximación lineal del contorno a partir de los puntos de curvatura cero; (e) puntos de curvatura extrema sobre el contorno suavizado; (f) aproximación lineal del contorno a partir de los puntos de curvatura extrema.

donde N_1 y N_2 indican el número de puntos de los contornos 1 y 2, respectivamente.

Los puntos críticos de cada uno de los contornos son numerados en función de su posición original. La figura 11.7 muestra el proceso de obtención de los puntos críticos de dos contornos.

El uso de esta técnica para generar los puntos críticos está justificada por dos motivos:

- El uso de los puntos de curvatura extrema y de los puntos equivalentes permite obtener una representación más característica de los contornos originales.
- El número total de puntos críticos de cada contorno es el mismo, condición exigida por el algoritmo de comparación.

11.4 Algoritmo de Gu y Tjahjadi

Una vez obtenidos los puntos críticos de los dos contornos, se procede a la aplicación del algoritmo de comparación, el cual está basado en el algoritmo de mínimos cuadrados propuesto por Gu y Tjahjadi (2.000).

Si se utilizan las coordenadas polares de los puntos respecto del centroide de su contorno, se tiene que

$$P_1(i) = (r_1(i), \alpha_1(i)) \quad (11.12)$$

y

$$P_2(i) = (r_2(i), \alpha_2(i)) \quad (11.13)$$

son los puntos críticos de cada uno de los contornos, donde $r_j(i)$ y $\alpha_j(i)$ representan el radio y ángulo del punto crítico i -ésimo del contorno $j \in \{1, 2\}$.

El objetivo es minimizar la siguiente función

$$\varepsilon^2(\text{contorno}_1, \text{contorno}_2(\beta, \theta)) = \frac{1}{N} \sum_{i=0}^{N-1} d(P_1(i), P_{2(\beta, \theta)}(i))^2 \quad (11.14)$$

donde

- d denota la distancia euclídea entre dos puntos,
- $P_{2(\beta, \theta)}(i)$ representa a un punto crítico del segundo contorno que ha sido rotado un ángulo θ y su radio ha sido escalado con una magnitud β , es decir,

$$P_{2(\beta, \theta)}(i) = (\beta * r_2(i), \alpha_2(i) + \theta) \quad (11.15)$$

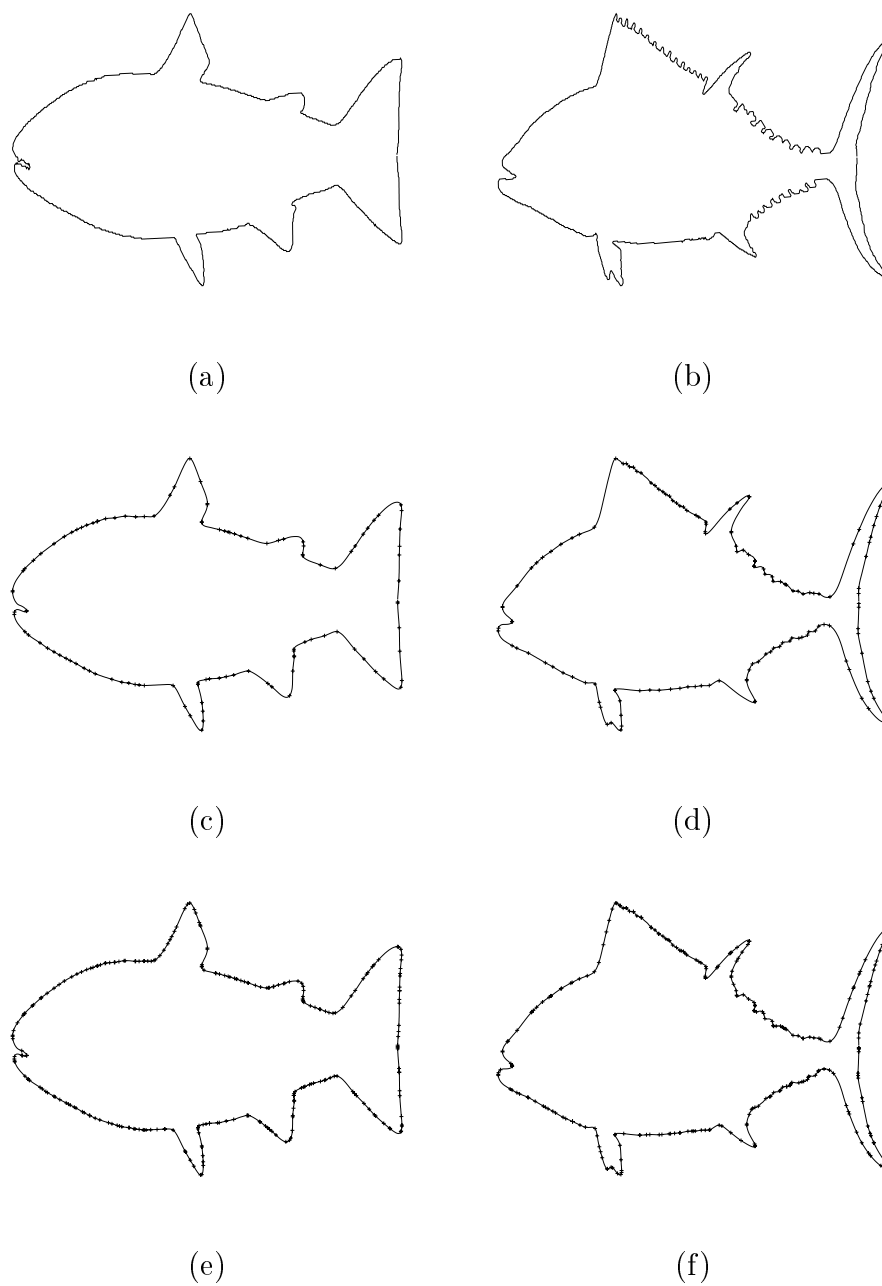


Figura 11.7: Obtención de los puntos críticos de dos contornos ($\sigma = 5$): (a) contorno original 1; (b) contorno original 2; (c) puntos de curvatura extrema sobre el contorno suavizado 1; (d) puntos de curvatura extrema sobre el contorno suavizado 2; (e) puntos de curvatura extrema y equivalentes sobre el contorno suavizado 1; (f) puntos de curvatura extrema y equivalentes sobre el contorno suavizado 2.

- y N es el número de puntos críticos de cada uno de los contornos.

Se ha de indicar que la función 11.14 considera que los contornos han sido trasladados de forma que sus centroides coincidan y, por tanto, sólo se ha de determinar cuánto se ha de escalar y rotar el segundo contorno para que se ajuste al primer contorno.

La función 11.14 es equivalente a

$$\varepsilon^2(\text{contorno}_1, \text{contorno}_2(\beta, \theta)) = \frac{1}{N} \sum_{i=0}^{N-1} (dx(i)^2 + dy(i)^2) \quad (11.16)$$

donde

$$dx(i) = r_1(i)\cos(\alpha_1(i)) - \beta r_2(i)\cos(\alpha_2(i) + \theta) \quad (11.17)$$

y

$$dy(i) = r_1(i)\sin(\alpha_1(i)) - \beta r_2(i)\sin(\alpha_2(i) + \theta) \quad (11.18)$$

Se ha hecho uso del método del gradiente para minimizar la función 11.16 con respecto a β y θ , es decir, estableciendo

$$\frac{\partial \varepsilon^2}{\partial \beta} = 0 \quad (11.19)$$

y

$$\frac{\partial \varepsilon^2}{\partial \theta} = 0 \quad (11.20)$$

Los parámetros de rotación y escala en la iteración $(k+1)$ -ésima son actualizados a partir de la iteración número k usando el método del descenso por la máxima pendiente:

$$\beta_{k+1} = \beta_k - \mu_1 \nabla_{\beta_k} \quad (11.21)$$

$$\theta_{k+1} = \theta_k - \mu_2 \nabla_{\theta_k} \quad (11.22)$$

donde

$$\nabla_{\beta_k} = \frac{\partial \varepsilon_k^2}{\partial \beta_k} \quad (11.23)$$

$$\nabla_{\theta_k} = \frac{\partial \varepsilon_k^2}{\partial \theta_k} \quad (11.24)$$

y μ_1 y μ_2 son dos constantes positivas pequeñas que controlan la velocidad de convergencia y la estabilidad del método.

Para la iteración inicial, el valor de β_0 se obtiene a partir del factor de escala inicial entre los puntos críticos de los contornos 1 y 2, es decir,

$$\beta_0 = \frac{\sum_{i=0}^{N-1} r_1(i)}{\sum_{i=0}^{N-1} r_2(i)} \quad (11.25)$$

y

$$\theta_0 = 0 \quad (11.26)$$

La utilización de este valor de θ_0 es debido a que los contornos han sido rotados previamente para que sus ejes de mínima inercia sean horizontales, teniendo, por tanto, la misma orientación (sección 11.2).

Las ecuaciones 11.23 y 11.24 son equivalentes a

$$\begin{aligned} \nabla_{\beta_k} &= \frac{\partial \varepsilon_k^2}{\partial \beta_k} \\ &= \frac{\partial}{\partial \beta_k} \left(\frac{1}{N} \sum_{i=0}^{N-1} (dx_k(i)^2 + dy_k(i)^2) \right) \\ &= \frac{-2}{N} \sum_{i=0}^{N-1} (dx_k(i)r_2(i)\cos(\alpha_2(i) + \theta_k) + dy_k(i)r_2(i)\sin(\alpha_2(i) + \theta_k)) \end{aligned} \quad (11.27)$$

y

$$\begin{aligned} \nabla_{\theta_k} &= \frac{\partial \varepsilon_k^2}{\partial \theta_k} \\ &= \frac{\partial}{\partial \theta_k} \left(\frac{1}{N} \sum_{i=0}^{N-1} (dx_k(i)^2 + dy_k(i)^2) \right) \\ &= \frac{2\beta}{N} \sum_{i=0}^{N-1} (dx_k(i)r_2(i)\sin(\alpha_2(i) + \theta_k) - dy_k(i)r_2(i)\cos(\alpha_2(i) + \theta_k)) \end{aligned} \quad (11.28)$$

El proceso de minimización de la función de comparación concluye cuando

$$\varepsilon^2(\text{contorno}_1, \text{contorno}_2(\beta_k, \theta_k)) = 0 \quad (11.29)$$

o

$$|\varepsilon^2(\text{contorno}_1, \text{contorno}_2(\beta_{k+1}, \theta_{k+1})) - \varepsilon^2(\text{contorno}_1, \text{contorno}_2(\beta_k, \theta_k))| < \epsilon \quad (11.30)$$

donde ϵ es una cota de error, o después de haber realizado un número predeterminado de iteraciones.

Para aplicar el algoritmo de comparación de contornos desarrollado se siguen las siguientes fases:

1. Parametrización de los contornos (sección 11.2): se obtienen las dos curvas parametrizadas de cada uno de los contornos. Cada curva parametrizada de un contorno debe ser comparada con cada una de las curvas parametrizadas del otro contorno.

2. Obtención de los puntos críticos a partir de las curvas parametrizadas de cada contorno. Puesto que el cálculo de los puntos críticos se basa en el algoritmo de Mokhtarian y Mackworth, se utilizarán dos valores de σ (σ_1 y σ_2) para suavizar cada una de las curvas parametrizadas correspondientes a los contornos de los objetos a comparar.
3. Uso de los puntos críticos para minimizar la función del algoritmo de comparación.

Este proceso se ha de aplicar repetidas veces a cada una de las curva parametrizadas de cada contorno, modificando los valores de los parámetros de suavización (σ_1 y σ_2) de cada curva. El valor que se obtenga como mínimo de la función al final de este proceso reflejará la discrepancia del segundo contorno respecto del primero.

11.5 Mejora del algoritmo de comparación

El uso de la función de comparación Gu y Tjahjadi (2.000) (función 11.14) presenta dos inconvenientes:

- La función no es simétrica, puesto que el segundo contorno ha de ser escalado y rotado para ajustarse al tamaño del primer contorno. La falta de simetría de la función impide que se pueda definir una métrica a partir de ella.
- El valor de la función no está normalizado, ya que depende del tamaño del primer contorno.

Para superar estas dos dificultades, se proponen las siguientes modificaciones:

- Normalizar el tamaño de los contornos compuestos por los puntos críticos:

$$P'_j(i) = (r'_j(i), \alpha_j(i)) = \left(\frac{c}{rmax_j} r_j(i), \alpha_j(i)\right) \quad (11.31)$$

donde

$$P_j(i) = (r_j(i), \alpha_j(i)) \quad (11.32)$$

denota el punto crítico del contorno $j \in \{1, 2\}$, c es un número positivo y $rmax_j$ es el radio máximo de los puntos críticos del contorno j , es decir,

$$rmax_j = \text{máximo}\{r_j(i) | i \in \{1, \dots, N\}\} \quad (11.33)$$

Por tanto, a partir de contorno_j , se va a obtener $\text{contorno}'_j$ normalizado en el que todos los radios de los puntos van a estar comprendidos entre 0 y c .

- Utilizar como función de comparación alguna de las dos siguientes:
 - Función basada en el ajuste de los ángulos de los puntos críticos:

$$\varepsilon_1^2(\text{contorno}'_1(\theta_1), \text{contorno}'_2(\theta_2)) = \frac{100}{4c^2N} \sum_{i=0}^{N-1} d(P'_{1(\theta_1)}(i), P'_{2(\theta_2)}(i))^2 \quad (11.34)$$

donde

$$P'_{1(\theta_1)}(i) = (r'_1(i), \alpha_1(i) + \theta_1) \quad (11.35)$$

y

$$P'_{2(\theta_2)}(i) = (r'_2(i), \alpha_2(i) + \theta_2) \quad (11.36)$$

Esta función trata de rotar los contornos hasta encontrar el mejor ajuste entre ellos.

- Función basada en el ajuste de los ángulos y los radios de los puntos críticos:

$$\varepsilon_2^2(\text{contorno}'_1(\beta_1, \theta_1), \text{contorno}'_2(\beta_2, \theta_2)) = \frac{100}{4c^2N} \sum_{i=0}^{N-1} \frac{1}{2} (d(P'_1(i), P'_{2(\beta_2, \theta_2)}(i))^2 + d(P'_{1(\beta_1, \theta_1)}(i), P'_2(i))^2) \quad (11.37)$$

donde

$$P'_{1(\beta_1, \theta_1)}(i) = (\beta_1 * r'_1(i), \alpha_1(i) + \theta_1) \quad (11.38)$$

y

$$P'_{2(\beta_2, \theta_2)}(i) = (\beta_2 * r'_2(i), \alpha_2(i) + \theta_2) \quad (11.39)$$

Esta función trata de rotar y escalar el primer contorno para que se ajuste al segundo contorno y, de forma independiente, realizar la misma operación con el segundo contorno para ajustarlo al primero.

La aplicación de estas modificaciones permite obtener:

- Funciones simétricas, de forma que a partir de cada una de ellas se puede definir una métrica entre contornos.
- Funciones normalizadas: sus valores van a estar comprendidos en el intervalo $[0, 100]$. Se ha de tener en cuenta que, en el peor de los casos (puntos diametralmente opuestos), la distancia máxima entre dos puntos será igual a $2c$, ya que el radio máximo de un punto es c . Puesto que las funciones ε_1^2 y ε_2^2 utilizan la suma del cuadrado de las distancias entre los puntos, se ha introducido el factor de normalización $\frac{100}{4c^2N}$ en las ecuaciones 11.34 y 11.37 para que valor de cada una de ellas esté comprendido entre 0 y 100.

11.5.1 Función de comparación basada en el ajuste de los ángulos de los puntos críticos

La obtención de los parámetros de rotación de la función ε_1^2 (función 11.34) es análoga a la de la función original ε^2 descrita en la sección 11.4.

Si se define

$$d(P'_{1(\theta)}(i), P'_{2(\theta_2)}(i))^2 = dx(i)^2 + dy(i)^2 \quad (11.40)$$

donde

$$dx(i) = r'_1(i)\cos(\alpha_1(i) + \theta_1) - r'_2(i)\cos(\alpha_2(i) + \theta_2) \quad (11.41)$$

y

$$dy(i) = r'_1(i)\sin(\alpha_1(i) + \theta_1) - r'_2(i)\sin(\alpha_2(i) + \theta_2) \quad (11.42)$$

entonces la función 11.34 se puede reescribir como

$$\varepsilon_1^2(\text{contorno}'_1(\theta_1), \text{contorno}'_2(\theta_2)) = \frac{100}{4c^2N} \sum_{i=0}^{N-1} (dx(i)^2 + dy(i)^2) \quad (11.43)$$

Desarrollando un proceso similar al descrito en las ecuaciones 11.19 - 11.28, se obtienen las siguientes equivalencias:

$$\begin{aligned} \nabla_{\theta_{1_k}} &= \frac{\partial \varepsilon_{1_k}^2}{\partial \theta_{1_k}} \\ &= \frac{\partial}{\partial \theta_{1_k}} \left(\frac{100}{4c^2N} \sum_{i=0}^{N-1} (dx(i)^2 + dy(i)^2) \right) \\ &= \frac{200}{4c^2N} \sum_{i=0}^{N-1} (-dx_k(i)r'_1(i)\sin(\alpha_1(i) + \theta_{1_k}) + dy_k(i)r'_1(i)\cos(\alpha_1(i) + \theta_{1_k})) \end{aligned} \quad (11.44)$$

y

$$\begin{aligned} \nabla_{\theta_{2_k}} &= \frac{\partial \varepsilon_{2_k}^2}{\partial \theta_{2_k}} \\ &= \frac{\partial}{\partial \theta_{2_k}} \left(\frac{100}{4c^2N} \sum_{i=0}^{N-1} (dx(i)^2 + dy(i)^2) \right) \\ &= \frac{200}{4c^2N} \sum_{i=0}^{N-1} (dx_k(i)r'_2(i)\sin(\alpha_2(i) + \theta_{2_k}) - dy_k(i)r'_2(i)\cos(\alpha_2(i) + \theta_{2_k})) \end{aligned} \quad (11.45)$$

Los parámetros utilizados en la primera iteración son

$$\theta_{1_0} = 0 \quad (11.46)$$

y

$$\theta_{2_0} = 0.0 \quad (11.47)$$

debido a que los contornos han sido rotados previamente para que sus ejes de mínima inercia sean horizontales, teniendo, por tanto, la misma orientación (sección 11.2).

Los parámetros de rotación en la iteración $(k + 1)$ -ésima son actualizados a partir de la iteración número k usando el método del descenso por la máxima pendiente:

$$\theta_{1_{k+1}} = \theta_{1_k} - \mu_1 \nabla_{\theta_{1_k}} \quad (11.48)$$

$$\theta_{2_{k+1}} = \theta_{2_k} - \mu_2 \nabla_{\theta_{2_k}} \quad (11.49)$$

donde μ_1 y μ_2 son dos constantes positivas pequeñas que controlan la velocidad de convergencia y la estabilidad del método.

11.5.2 Función de comparación basada en el ajuste de los ángulos y los radios de los puntos críticos

El proceso mediante el cual se obtienen los parámetros de rotación y escala de la función ε_2 (función 11.37) es análogo a los procesos descritos previamente correspondientes a las funciones ε^2 (función 11.14) y ε_1^2 (función 11.34).

Si se establecen las siguientes definiciones

$$d(P'_1(i), P'_{2(\beta_2, \theta_2)}(i))^2 = dx_2(i)^2 + dy_2(i)^2 \quad (11.50)$$

y

$$d(P'_{1(\beta_1, \theta_1)}(i), P'_2(i))^2 = dx_1(i)^2 + dy_1(i)^2 \quad (11.51)$$

donde

$$dx_1(i) = r'_2(i) \cos(\alpha_2(i)) - \beta_1 r'_1(i) \cos(\alpha_1(i) + \theta_1) \quad (11.52)$$

$$dy_1(i) = r'_2(i) \sin(\alpha_2(i)) - \beta_1 r'_1(i) \sin(\alpha_1(i) + \theta_1) \quad (11.53)$$

y

$$dx_2(i) = r'_1(i) \cos(\alpha_1(i)) - \beta_2 r'_2(i) \cos(\alpha_2(i) + \theta_2) \quad (11.54)$$

$$dy_2(i) = r'_1(i) \sin(\alpha_1(i)) - \beta_2 r'_2(i) \sin(\alpha_2(i) + \theta_2) \quad (11.55)$$

entonces la función 11.37 es equivalente a

$$\begin{aligned} & \varepsilon_2^2(\text{contorno}'_1(\beta_1, \theta_1), \text{contorno}'_2(\beta_2, \theta_2)) = \\ & \frac{100}{4c^2 N} \sum_{i=0}^{N-1} \frac{1}{2} (dx_2(i)^2 + dy_2(i)^2 + dx_1(i)^2 + dy_1(i)^2) \end{aligned} \quad (11.56)$$

Realizando un proceso similar al descrito en las ecuaciones 11.19 - 11.28, se obtienen las siguientes equivalencias:

$$\begin{aligned}
 \nabla_{\beta_{1_k}} &= \frac{\partial \varepsilon_{2_k}^2}{\partial \beta_{1_k}} \\
 &= \frac{\partial}{\partial \beta_{1_k}} \left(\frac{100}{4c^2 N} \sum_{i=0}^{N-1} \frac{1}{2} (dx_2(i)^2 + dy_2(i)^2 + dx_1(i)^2 + dy_1(i)^2) \right) \\
 &= \frac{-100}{4c^2 N} \sum_{i=0}^{N-1} (dx_{1_k}(i)r'_1(i)\cos(\alpha_1(i) + \theta_{1_k}) + dy_{1_k}(i)r'_1(i)\sin(\alpha_1(i) + \theta_{1_k}))
 \end{aligned} \tag{11.57}$$

$$\begin{aligned}
 \nabla_{\theta_{1_k}} &= \frac{\partial \varepsilon_{2_k}^2}{\partial \theta_{1_k}} \\
 &= \frac{\partial}{\partial \theta_{1_k}} \left(\frac{100}{4c^2 N} \sum_{i=0}^{N-1} \frac{1}{2} (dx_2(i)^2 + dy_2(i)^2 + dx_1(i)^2 + dy_1(i)^2) \right) \\
 &= \frac{100\beta_1}{4c^2 N} \sum_{i=0}^{N-1} (dx_{1_k}(i)r'_1(i)\sin(\alpha_1(i) + \theta_{1_k}) - dy_{1_k}(i)r'_1(i)\cos(\alpha_1(i) + \theta_{1_k}))
 \end{aligned} \tag{11.58}$$

y

$$\begin{aligned}
 \nabla_{\beta_{2_k}} &= \frac{\partial \varepsilon_{2_k}^2}{\partial \beta_{2_k}} \\
 &= \frac{\partial}{\partial \beta_{2_k}} \left(\frac{100}{4c^2 N} \sum_{i=0}^{N-1} \frac{1}{2} (dx_2(i)^2 + dy_2(i)^2 + dx_1(i)^2 + dy_1(i)^2) \right) \\
 &= \frac{-100}{4c^2 N} \sum_{i=0}^{N-1} (dx_{2_k}(i)r'_2(i)\cos(\alpha_2(i) + \theta_{2_k}) + dy_{2_k}(i)r'_2(i)\sin(\alpha_2(i) + \theta_{2_k}))
 \end{aligned} \tag{11.59}$$

$$\begin{aligned}
 \nabla_{\theta_{2_k}} &= \frac{\partial \varepsilon_{2_k}^2}{\partial \theta_{2_k}} \\
 &= \frac{\partial}{\partial \theta_{2_k}} \left(\frac{100}{4c^2 N} \sum_{i=0}^{N-1} \frac{1}{2} (dx_2(i)^2 + dy_2(i)^2 + dx_1(i)^2 + dy_1(i)^2) \right) \\
 &= \frac{100\beta_2}{4c^2 N} \sum_{i=0}^{N-1} (dx_{2_k}(i)r'_2(i)\sin(\alpha_2(i) + \theta_{2_k}) - dy_{2_k}(i)r'_2(i)\cos(\alpha_2(i) + \theta_{2_k}))
 \end{aligned} \tag{11.60}$$

Puesto que se han normalizado previamente los contornos formados por los puntos críticos, los valores utilizados en la iteración inicial son

$$\beta_{1_0} = \beta_{2_0} = 1.0 \tag{11.61}$$

mientras que

$$\theta_{1_0} = 0.0 \tag{11.62}$$

y

$$\theta_{2_0} = 0.0 \tag{11.63}$$

debido a que los contornos han sido rotados previamente para que sus ejes de mínima inercia sean horizontales, teniendo, por tanto, la misma orientación (sección 11.2).

Los parámetros de rotación y escala en la iteración $(k + 1)$ -ésima son actualizados a partir de la iteración número k usando el método del descenso por la máxima pendiente:

$$\beta_{1_{k+1}} = \beta_{1_k} - \mu_1 \nabla_{\beta_{1_k}} \tag{11.64}$$

$$\theta_{1_{k+1}} = \theta_{1_k} - \mu_2 \nabla_{\theta_{1_k}} \tag{11.65}$$

$$\beta_{2_{k+1}} = \beta_{2_k} - \mu_3 \nabla_{\beta_{2_k}} \tag{11.66}$$

$$\theta_{2_{k+1}} = \theta_{2_k} - \mu_4 \nabla_{\theta_{2_k}} \tag{11.67}$$

donde $\mu_1, \mu_2, \mu_3, \mu_4$ son constantes positivas pequeñas que controlan la velocidad de convergencia y la estabilidad del método.

Capítulo 12

Análisis de la metodología de reconocimiento

12.1 Introducción

La metodología de reconocimiento de objetos basada en el análisis de sus contornos es resumida en la sección 12.2.

En la sección 12.3, se describen diferentes pruebas realizadas para comprobar el funcionamiento de la metodología, analizando especialmente los resultados obtenidos por las funciones de comparación de contornos descritas en la sección 11.5.

Por último, la sección 12.4 describe los resultados obtenidos al aplicar esta metodología de reconocimiento.

12.2 Fases de la metodología de reconocimiento

La metodología diseñada para comparar dos objetos utilizando sus contornos se puede resumir en las siguientes fases:

1. Preprocesamiento de las imágenes que contienen los objetos.

Si se considera necesario, se puede aplicar un algoritmo de suavización de imágenes en color para atenuar la influencia del ruido (sección 2.5). Esta fase es opcional.

2. Detección de bordes y umbralización.

El algoritmo elegido para la detección de los bordes en imágenes en color ha sido el propuesto por Lee y Cok (1991), debido a que ha sido, junto con el algoritmo de Di Zenzo (1986), el método que ha tenido un mejor rendimiento en las pruebas de evaluación de detectores de bordes descritas en el capítulo 8.

Para la identificación de los puntos de los bordes, se ha optado por el método de umbralización porcentual (subsección 3.3.5) (Pratt 1978; Sonka et al. 1993).

3. Extracción de los contornos exteriores.

Se ha de utilizar la metodología descrita en el capítulo 10 para extraer el contorno exterior del objeto contenido en cada una de las imágenes.

4. Parametrización de los contornos.

Para evitar las dificultades provocadas por las rotaciones, se han de parametrizar los contornos utilizando el método descrito en la sección 11.2, lo que provocará una rotación de los contornos para que sus ejes de mínima inercia sean horizontales.

5. Obtención de los puntos críticos de cada contorno.

Para comparar dos contornos, se han elegido como descriptores los siguientes puntos ya descritos en la subsección 11.3.2:

- Puntos de curvatura extrema (máxima o mínima) de cada contorno.
- Puntos de un contorno equivalentes a puntos de curvatura extrema del otro contorno.

Los puntos de curvatura extrema han sido calculados mediante el algoritmo de Mokhtarian y Mackworth (1.992) descrito en la sección 11.3.1 (Mokhtarian y Mackworth 1992; Mokhtarian 1995).

El algoritmo de Mokhtarian y Mackworth utiliza un parámetro σ para convolucionar cada contorno con el filtro de la función gaussiana. Si se desea, se pueden utilizar diferentes valores de σ para cada uno de los contornos, pudiendo efectuar de este modo un análisis comparativo en el espacio de escalas.

6. Normalización del tamaño de los contornos.

Se ha de normalizar el tamaño de los radios de los puntos característicos de los contornos para que los valores de las funciones de comparación puedan también estar normalizados (sección 11.5).

7. Minimización de la función de comparación de contornos.

Se ha de utilizar una de las dos funciones que mejoran la comparación de los contornos y que han sido descritas en la sección 11.5:

- Función basada en el ajuste de los ángulos de los puntos críticos (ecuación 11.34):
 - Esta función calcula la suma del cuadrado de las distancias entre los puntos críticos de los dos contornos.
 - Esta función posee dos parámetros θ_1 y θ_2 que representan, respectivamente, los ángulos utilizados para rotar los puntos críticos de cada contorno.
 - Al minimizar la función, se obtienen los valores θ_{1_k} y θ_{2_k} , correspondientes a los ángulos que han de rotarse los puntos críticos de cada contorno, para que la suma del cuadrado de las distancias entre los puntos críticos sea mínima.
- Función basada en el ajuste de los ángulos y los radios de los puntos críticos (ecuación 11.37):
 - Esta función calcula la suma de la media aritmética del cuadrado de las distancias entre los puntos críticos de los dos contornos.
 - Esta función posee cuatro parámetros: θ_1 y β_1 , θ_2 y β_2 . Los parámetros θ_i y β_i indican, respectivamente, el ángulo de rotación y el factor de escala que se ha aplicado a los puntos críticos del contorno i ($i \in \{1, 2\}$).
 - Al minimizar la función, se obtienen los valores θ_{1_k} , β_{1_k} , θ_{2_k} y β_{2_k} , correspondientes a los ángulos y las escalas que han de aplicarse a los puntos críticos de cada contorno, para que la suma de la media aritmética del cuadrado de las distancias entre los puntos críticos sea mínima.

12.3 Pruebas de la metodología de reconocimiento

Se han diseñado y realizado una serie de pruebas para comprobar el funcionamiento de la metodología de reconocimiento de objetos propuesta. Estas pruebas han sido las siguientes:

- Análisis de las funciones de comparación de contornos.
- Estudio de la influencia de las rotaciones y cambios de escala.
- Pruebas con imágenes reales.

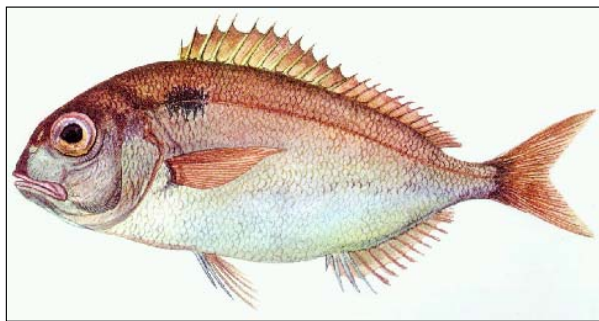
En las siguientes secciones se describen las características y los resultados de las pruebas realizadas.

12.3.1 Análisis de las funciones de comparación

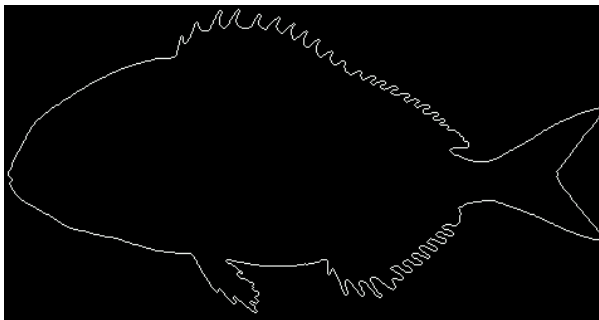
La primera prueba diseñada trata de comprobar si los valores obtenidos por las dos funciones de comparación de contornos están o no correlacionados.

Para realizar esta prueba, se utilizaron 36 imágenes de peces como la que se muestra en la figura 12.1(a) y cuyas características son

- Imágenes en color en el espacio RGB (formato *kdf* de *Khoros 2.2*).
- Número de niveles en cada plano de color: 256.
- Tipo de dato de los niveles de color: real de punto flotante (*float*).
- Altura: 256.
- Anchura: variable.



(a)



(b)

Figura 12.1: Ejemplo de imagen en color utilizada en la primera prueba de comparación: (a) besugo; (b) contorno de (a).

Los parámetros utilizados en las distintas fases del proceso de comparación de los contornos fueron:

- Los valores de σ_1 y σ_2 , utilizados por el algoritmo de Mokhtarian y Mackworth (1.992) para obtener los puntos de curvatura extrema de cada uno de los contornos, variaron entre 1 y 5 con incrementos de 0.5.

- Los valores de los multiplicadores de las funciones de comparación fueron:

- Multiplicadores de la función basada en el ajuste de los ángulos (ecuaciones 11.48 y 11.49):

$$\mu_1 = \mu_2 = 0.0001 \quad (12.1)$$

- Multiplicadores de la función basada en el ajuste de los ángulos y los radios (ecuaciones 11.64 - 11.67):

$$\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.0001 \quad (12.2)$$

- Criterios de parada del proceso de minimización:

- Número máximo de iteraciones

$$N = 1000 \quad (12.3)$$

- Cota de error

$$\epsilon = 0.0001 \quad (12.4)$$

Para poder comparar los peces entre sí, se extrajeron en primer lugar los contornos exteriores de todos los peces (figura 12.1(b)).

El contorno de cada pez fue comparado con cada uno de los 35 contornos de los demás peces. Debido a que las funciones de comparación son simétricas, el número de pruebas realizadas se redujo a la mitad, es decir, se hicieron 630 comparaciones con cada una de las funciones.

Al calcular el valor mínimo $\varepsilon_{1_k}^2$, correspondiente a la función de comparación basada en el ajuste de los ángulos de los puntos críticos, se obtuvieron los siguientes valores auxiliares:

- k : número de iteración en la que se alcanzó el valor mínimo.
- σ_1 : valor de σ con el que se aplicó el filtro de la gaussiana al primer contorno para obtener el valor mínimo.
- σ_2 : valor de σ con el que se aplicó el filtro de la gaussiana al segundo contorno para obtener el valor mínimo.

- θ_{1_k} : ángulo que ha de rotarse el primer contorno para alcanzar el valor mínimo.
- θ_{2_k} : ángulo que ha de rotarse el segundo contorno para alcanzar el valor mínimo.
- N_1 : número de puntos del primer contorno.
- N_2 : número de puntos del segundo contorno.
- C_1 : número de puntos críticos del primer contorno obtenidos con el valor σ_1 .
- C_2 : número de puntos críticos del segundo contorno obtenidos con el valor σ_2 .

Los valores θ_{1_k} y θ_{2_k} representan las rotaciones que han de efectuar los contornos para ajustarse entre sí.

Valores auxiliares similares se obtuvieron al calcular el valor mínimo $\varepsilon_{e_k}^2$, correspondiente a la función basada en el ajuste de los ángulos y los radios de los puntos críticos. Además, también se obtuvieron los siguientes valores:

- β_{1_k} : factor de escala que se ha de aplicar al primer contorno para alcanzar el valor mínimo.
- β_{2_k} : factor de escala que se ha de aplicar al segundo contorno para alcanzar el valor mínimo.

En este caso, los valores θ_{1_k} , θ_{2_k} , β_{1_k} y β_{2_k} representan las rotaciones y los factores de escala que se han de utilizar para ajustar los dos contornos sí.

Las tablas 12.1 y 12.2 sólo muestran los resultados obtenidos por la función basada en el ajuste de los ángulos al comparar un besugo con el resto de los peces. Análogamente, los valores obtenidos por el besugo al utilizar la función basada en el ajuste de los ángulos y los radios de los puntos críticos se muestran en las tablas 12.3 y 12.4.

Si se analizan los resultados generados por todos los peces comparados, se pueden extraer las siguientes conclusiones:

- Las dos funciones de comparación están correlacionadas, como refleja el coeficiente de Pearson:

$$Pearson(\varepsilon_{1_k}^2, \varepsilon_{2_k}^2) = 0.998 \quad (12.5)$$

Esta correlación es significativa al nivel 0.01 (bilateral).

Este es el resultado más importante que se puede deducir de las pruebas realizadas, ya que, para comparar dos contornos normalizados, sólo hay que realizar el ajuste de los ángulos de los puntos críticos.

Por tanto, sólo es necesario calcular los valores de θ_{1_k} y θ_{2_k} que indican, respectivamente, cuánto ha de rotarse cada uno de los contornos, en vez de calcular los cuatro parámetros que requiere la función basada en el ajuste de los ángulos y los radios de los puntos críticos. Como consecuencia inmediata, la complejidad computacional se reduce a la mitad.

- Los ángulos de rotación θ_{1_k} y θ_{2_k} obtenidos al minimizar la función basada en el ajuste de los ángulos también están correlacionados

$$\text{Pearson}(\theta_{1_k}, \theta_{2_k}) = -1.000 \quad (12.6)$$

Esta correlación es significativa al nivel 0.01 (bilateral).

El valor -1.000 refleja que los ángulos son opuestos entre sí. Este resultado es lógico, ya que si los contornos no cambian de escala entonces sólo es necesario que giren en sentidos opuestos la misma magnitud angular para que se produzca el ajuste entre ellos.

Unificando, los dos resultados obtenidos, sólo es necesario calcular el ángulo que ha de rotarse un contorno, ya que el ángulo correspondiente al otro contorno será el opuesto del anterior. Por tanto, se vuelve a reducir la complejidad computacional a la mitad.

- Considérense los estadísticos descriptivos del parámetro θ_{1_k} , correspondiente al ángulo de rotación del primer contorno donde la función basada en el ajuste de los ángulos alcanza el valor mínimo:

<i>Media</i>	=	$-3.26e - 19$
<i>Desviación típica</i>	=	$1.461e - 02$
<i>Mínimo</i>	=	-0.069927
<i>Máximo</i>	=	0.069927
<i>Rango</i>	=	0.139854

Estos descriptores indican que el ángulo θ_{1_k} está muy próximo a cero, es decir, el contorno apenas ha de rotarse para alcanzar el valor mínimo.

Por consiguiente, la utilización del eje de mínima inercia para parametrizar cada contorno y del ángulo del primer punto para realizar la rotación inicial (sección 11.2) son suficientes para conseguir una buena aproximación al valor mínimo de la función de comparación.

Este resultado indica que se puede prescindir del proceso de minimización de la función si sólo se desea obtener una buena aproximación al valor mínimo.

Tabla 12.1: Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación de ajuste de los ángulos (primera parte).

	Pez 1	Pez 2	$\epsilon_{1_k}^2$	σ_1	σ_2	k	θ_{1_k}	θ_{2_k}	N_1	N_2	C_1	C_2
Besugo	Abadeama	0.590752	5.00	2.50	198	-0.018892	0.018892	1681	1497	142	214	
Besugo	Albacora	0.711442	2.50	1.50	2	0.000076	-0.000076	1681	1808	250	438	
Besugo	Arenque1	0.659442	1.50	1.00	54	-0.004154	0.004154	1681	1575	398	552	
Besugo	Arenque2	0.791476	4.00	1.00	88	-0.007057	0.007057	1681	1772	164	574	
Besugo	Atún	0.660470	2.50	3.50	2	-0.000015	0.000015	1681	1889	250	204	
Besugo	Bacalao	0.775024	1.00	3.50	132	-0.011791	0.011791	1681	1695	608	168	
Besugo	Barborio	0.879183	4.50	3.50	2	0.000112	-0.000112	1681	1649	152	152	
Besugo	Bonito-sur	0.739248	1.50	3.50	2	0.000070	-0.000070	1681	1798	396	182	
Besugo	Boquerón	1.034270	2.00	1.50	2	-0.000128	0.000128	1681	2150	302	516	
Besugo	Breca	0.221540	2.50	4.50	2	-0.000046	0.000046	1681	1503	250	120	
Besugo	Caballa	0.701313	1.50	3.50	216	-0.021798	0.021798	1681	1725	398	182	
Besugo	Carpa	0.343195	5.00	2.50	2	-0.000091	0.000091	1681	1330	142	190	
Besugo	Corvina	0.541423	2.50	1.00	32	-0.002383	0.002383	1681	1565	250	582	
Besugo	Dorada	0.655983	3.50	4.50	250	-0.027143	0.027143	1681	1367	184	112	
Besugo	Esturion	0.917736	5.00	3.50	2	-0.000018	0.000018	1681	2435	142	240	
Besugo	Faneca	0.704305	2.50	3.50	357	-0.048583	0.048583	1681	1314	250	130	
Besugo	Gallo	0.723743	2.50	5.00	2	0.000043	-0.000043	1681	1418	250	102	
Besugo	Hurta-hembra	0.725365	2.50	3.50	2	-0.000044	0.000044	1681	1081	250	116	

Tabla 12.2: Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación de ajuste de los ángulos (segunda parte).

Pez 1	Pez 2	ε_1^2	σ_1	σ_2	k	θ_{1_k}	θ_{2_k}	N_1	N_2	C_1	C_2
Besugo	Hurta-macho	1.141512	4.00	4.00	142	0.012332	-0.012332	1681	1813	164	158
Besugo	Jurel	0.544881	5.00	2.50	2	-0.000119	0.000119	1681	1545	142	248
Besugo	Lubina	0.666461	2.50	3.50	2	0.000138	-0.000138	1681	1914	250	204
Besugo	Melva	0.751655	4.00	2.00	2	0.000029	-0.000029	1681	1962	164	398
Besugo	Merluza	0.723048	4.50	4.50	15	0.001076	-0.001076	1681	2447	152	234
Besugo	Mero	0.288595	5.00	4.00	2	0.000102	-0.000102	1681	1495	142	138
Besugo	Perca	0.290839	5.00	5.00	89	-0.007045	0.007045	1681	1399	142	88
Besugo	Pez-espada	0.888842	4.00	3.50	2	-0.000041	0.000041	1681	2086	164	222
Besugo	Pez-limón	0.800586	5.00	2.00	10	-0.000714	0.000714	1681	1632	142	282
Besugo	Raya1	0.831744	2.00	3.50	22	-0.001604	0.001604	1681	918	302	84
Besugo	Rodaballo	1.028874	1.00	4.00	2	-0.000133	0.000133	1681	1037	608	90
Besugo	Sama	0.202248	5.00	3.00	2	-0.000012	0.000012	1681	1469	142	170
Besugo	Sardina	0.625872	2.00	2.50	2	-0.000013	0.000013	1681	1680	302	270
Besugo	Tenca	0.345757	4.50	2.00	2	-0.000010	0.000010	1681	1660	152	322
Besugo	Trucha-mar1	0.727376	4.50	2.00	2	-0.000080	0.000080	1681	1734	152	326
Besugo	Trucha-mar2	0.697346	2.00	1.00	141	-0.012370	0.012370	1681	1741	302	580
Besugo	Trucha-riol	0.626496	5.00	2.00	2	-0.000095	0.000095	1681	1823	142	328

Tabla 12.3: Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación basada en el ajuste de los ángulos y los radios de los puntos críticos (primera parte).

Pez 1	Pez 2	ε_{2k}^2	σ_1	σ_2	k	θ_{1k}	θ_{2k}	β_{1k}	β_{2k}	N_1	N_2	C_1	C_2
Besugo	Abadeama	0.611250	5.00	2.50	99	-0.006007	0.005972	1.003155	0.990931	1681	1497	142	214
Besugo	Albacora	0.711427	2.50	1.50	2	0.000038	-0.000038	0.999777	0.999881	1681	1808	250	438
Besugo	Arenque1	0.662137	4.00	1.00	2	-0.000074	0.000074	0.999949	0.999919	1681	1575	164	552
Besugo	Arenque2	0.802668	4.00	1.00	2	-0.000091	0.000091	0.999909	0.999930	1681	1772	164	574
Besugo	Atún	0.660448	2.50	3.50	2	-0.000008	0.000008	0.999918	0.999950	1681	1889	250	204
Besugo	Bacalao	0.769375	2.50	2.50	99	-0.001866	0.001853	1.003574	0.989054	1681	1695	250	240
Besugo	Barborio	0.865857	4.50	3.50	99	0.002688	-0.002671	1.002337	0.989325	1681	1649	152	152
Besugo	Bonito-sur	0.739239	1.50	3.50	2	0.000035	-0.000035	0.999931	0.999921	1681	1798	396	182
Besugo	Boquerón	1.034276	2.00	1.50	2	-0.000064	0.000064	0.999915	0.999878	1681	2150	302	516
Besugo	Breca	0.221540	2.50	4.50	2	-0.000023	0.000023	1.000013	0.999943	1681	1503	250	120
Besugo	Caballa	0.735584	1.50	3.50	99	-0.006608	0.006607	0.996694	0.996246	1681	1725	398	182
Besugo	Carpa	0.343166	5.00	2.50	2	-0.000045	0.000045	1.000070	0.999861	1681	1330	142	190
Besugo	Corvina	0.539024	2.50	5.00	99	-0.002892	0.002874	1.004030	0.990790	1681	1565	250	110
Besugo	Dorada	0.687440	3.50	4.50	99	-0.007552	0.007485	1.005858	0.987463	1681	1367	184	112
Besugo	Esturion	0.917695	5.00	3.50	2	-0.000009	0.000009	0.999902	0.999915	1681	2435	142	240
Besugo	Faneca	0.734975	2.50	3.50	2	-0.000080	0.000080	0.999995	0.999859	1681	1314	250	130
Besugo	Gallo	0.687877	2.50	5.00	99	0.001023	-0.001010	1.009591	0.983744	1681	1418	250	102
besugo	Hurta-hembra	0.686640	2.50	3.50	99	-0.001041	0.001028	1.010124	0.983200	1681	1081	250	116

Tabla 12.4: Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación basada en el ajuste de ángulos y los radios de los puntos críticos (segunda parte).

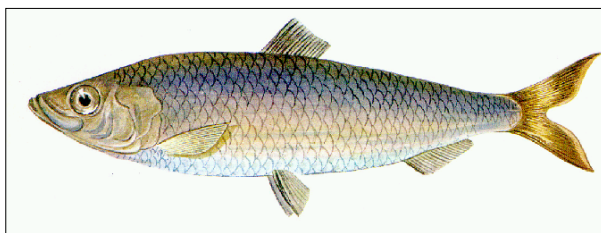
Pez 1	Pez 2	$\varepsilon_{2_k}^2$	σ_1	σ_2	k	θ_{1_k}	θ_{2_k}	β_{1_k}	β_{2_k}	N_1	N_2	C_1	C_2
Besugo	Hurta-macho	1.143576	4.00	4.00	99	0.005047	-0.005015	1.000956	0.988022	1681	1813	164	158
Besugo	Jurel	0.544881	5.00	2.50	2	-0.000060	0.000060	1.000033	0.999858	1681	1545	142	248
Besugo	Lubina	0.666497	2.50	3.50	2	0.000069	-0.000069	0.999985	0.999882	1681	1914	250	204
Besugo	Melva	0.751630	4.00	2.00	2	0.000015	-0.000015	0.999921	0.999929	1681	1962	164	398
Besugo	Merluza	0.724412	4.50	4.50	2	0.000073	-0.000073	0.999855	1.000000	1681	2447	152	234
Besugo	Mero	0.288601	5.00	4.00	2	0.000051	-0.000051	1.000047	0.999895	1681	1495	142	138
Besugo	Perca	0.301847	5.00	5.00	2	-0.000089	0.000089	1.000057	0.999882	1681	1399	142	88
Besugo	Pez-espada	0.888783	4.00	3.50	2	-0.000021	0.000021	0.999838	0.999984	1681	2086	164	222
Besugo	Pez-limon	0.780212	5.00	2.00	99	-0.003466	0.003436	1.005180	0.987268	1681	1632	142	282
Besugo	Rayal	0.812104	2.00	3.50	99	-0.003598	0.003567	1.005101	0.987083	1681	918	302	84
Besugo	Rodaballo	0.947341	1.00	4.00	99	-0.003165	0.003107	1.014755	0.975967	1681	1037	608	90
Besugo	Sama	0.202239	5.00	3.00	2	-0.000006	0.000006	1.000019	0.999941	1681	1469	142	170
Besugo	Sardina	0.625852	2.00	2.50	2	-0.000006	0.000006	0.999948	0.999927	1681	1680	302	270
Besugo	Tenca	0.345751	4.50	2.00	2	-0.000005	0.000005	0.999974	0.999957	1681	1660	152	322
Besugo	Trucha-mar1	0.727336	4.50	2.00	2	-0.000040	0.000040	1.000014	0.999841	1681	1734	152	326
Besugo	Trucha-mar2	0.717110	4.00	1.00	2	-0.000096	0.000096	0.999914	0.999943	1681	1741	164	580
Besugo	Trucha-rio1	0.626479	5.00	2.00	2	-0.000047	0.000047	1.000016	0.999858	1681	1823	142	328

12.3.2 Estudio de la influencia de la orientación y el tamaño iniciales de los objetos

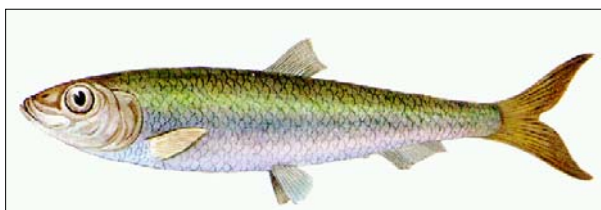
La segunda prueba diseñada pretendía comprobar la influencia de la orientación y el tamaño iniciales de los objetos sobre la metodología de comparación.

Aunque las funciones de comparación propuestas son invariantes a cambios de escala o rotaciones de los contornos, los algoritmos de detección de bordes, umbralización de imágenes y extracción del contorno exterior no son capaces de obtener los mismos contornos de dos objetos que difieran en la escala o en la orientación.

Debido a ello, se ha analizado el rendimiento de la función de comparación basada en el ajuste de los ángulos al comparar contornos de objetos rotados y escalados obtenidos por los algoritmos de detección de bordes, umbralización imágenes y extracción del contorno exterior.



(a)



(b)

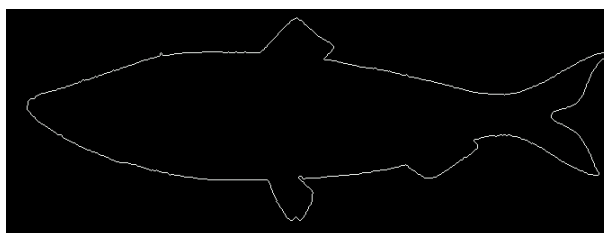
Figura 12.2: Dos de las imágenes en color utilizadas en la segunda prueba de comparación de contornos: (a) arenque 1; (b) arenque 2.

Se eligieron inicialmente dos imágenes que contenían cada una de ellas un arenque (figura 12.2) y cuyas características eran

- Imágenes en color en el espacio RGB (formato *kdf* de *Khoros 2.2*).
- Número de niveles en cada plano de color: 256.

- Tipo de dato de los niveles de color: real de punto flotante (*float*).
- Altura: 256.
- Anchura de la imagen de arenque 1: 676.
- Anchura de la imagen de arenque 2: 744.

Estas imágenes fueron rotadas 0° , 30° , 45° y 90° y escaladas por un factor igual a 0.5, 1.0 y 1.5. Por tanto, se generaron 12 para cada uno de los arenques. Todos los peces contenidos en estas 24 imágenes fueron comparados entre sí.



(a)



(b)

Figura 12.3: Contornos exteriores de los peces de la figura 12.2: (a) contorno de arenque 1; (b) contorno de arenque 2.

Los parámetros utilizados en las distintas fases del proceso de comparación de los contornos fueron los mismos que se indicaron en la primera prueba, salvo que el número máximo de iteraciones se fijó en 100.

La tabla 12.5 muestra los resultados obtenidos por la imagen correspondiente al arenque 1 rotada 30° y escalada con un factor igual a 0.5., donde G_i y E_i ($i \in \{1, 2\}$) son, respectivamente, los grados y la escala que se han aplicado a la imagen con el arenque i .

Los valores de la función basada en el ajuste de los ángulos obtenidos al comparar todos los peces entre sí han sido estudiados mediante la técnica del análisis de la varianza (ANOVA), que detecta las dependencias entre las variables

Tabla 12.5: Resultados obtenidos al comparar el arenque 1 rotado 30° y escalado 0.5 con los demás peces utilizando la función de comparación de ajuste de los ángulos.

	Pez 1	G_1	E_1	Pez 2	G_2	E_2	ε_{1k}^2	σ_1	σ_2	k	θ_{1k}	θ_{2k}	N_1	N_2	C_1	C_2
	Arenque 1	30°	.5	Arenque 1	0°	.5	.015167	4.00	5.00	2	-.000043	.000043	754	785	54	481
	Arenque 1	30°	.5	Arenque 1	0°	1.0	.019494	1.00	2.00	2	-.000041	.000041	754	1575	274	2961
	Arenque 1	30°	.5	Arenque 1	0°	1.5	.019494	1.00	2.00	2	-.000041	.000041	754	1575	274	2961
	Arenque 1	30°	.5	Arenque 1	30	1.0	.005729	1.00	4.50	2	-.000014	.000014	754	1534	274	1101
	Arenque 1	30°	.5	Arenque 1	30	1.5	.011745	1.50	3.50	2	-.000034	.000034	754	2384	176	2601
	Arenque 1	30°	.5	Arenque 1	45°	.5	.013530	4.50	4.00	2	-.000025	.000025	754	743	46	481
	Arenque 1	30°	.5	Arenque 1	45°	1.0	.020878	1.50	1.00	2	.000014	-.000014	754	1544	176	5441
	Arenque 1	30°	.5	Arenque 1	45°	1.5	.011879	1.50	5.00	2	-.000021	.000021	754	2384	176	1801
	Arenque 1	30°	.5	Arenque 1	90°	.5	.020941	4.00	5.00	2	-.000057	.000057	754	784	54	461
	Arenque 1	30°	.5	Arenque 1	90°	1.0	.024553	1.50	1.50	2	-.000023	.000023	754	1598	176	4041
	Arenque 1	30°	.5	Arenque 1	90°	1.5	.023226	1.50	4.50	2	-.000030	.000030	754	2454	176	1881
	Arenque 1	30°	.5	Arenque 2	0°	.5	.053250	4.00	2.00	2	-.000038	.000038	754	873	54	1681
	Arenque 1	30°	.5	Arenque 2	0°	1.0	.054381	1.50	1.00	2	-.000027	.000027	754	1772	176	5701
	Arenque 1	30°	.5	Arenque 2	0°	1.5	.052494	1.00	1.50	2	-.000021	.000021	754	2718	274	6461
	Arenque 1	30°	.5	Arenque 2	30	.5	.037106	2.50	3.50	2	-.000006	.000006	754	842	104	821
	Arenque 1	30°	.5	Arenque 2	30	1.0	.047385	1.50	5.00	2	-.000008	.000008	754	1710	176	1161
	Arenque 1	30°	.5	Arenque 2	30	1.5	.048592	1.00	2.00	2	-.000033	.000033	754	2629	274	5001
	Arenque 1	30°	.5	Arenque 2	45°	.5	.047277	1.00	2.50	2	-.000037	.000037	754	833	274	1201
	Arenque 1	30°	.5	Arenque 2	45°	1.0	.061397	1.50	1.00	2	-.000034	.000034	754	1662	176	5681
	Arenque 1	30°	.5	Arenque 2	45°	1.5	.058824	1.50	1.50	2	-.000028	.000028	754	2521	176	6301
	Arenque 1	30°	.5	Arenque 2	90°	.5	.049893	4.00	5.00	2	.000007	-.000007	754	872	54	541
	Arenque 1	30°	.5	Arenque 2	90°	1.0	.053147	1.50	1.50	2	-.000037	.000037	754	1765	176	4121
	Arenque 1	30°	.5	Arenque 2	90°	1.5	.116896	4.00	5.00	2	-.000019	.000019	754	2597	54	1861

y determina la significancia estadística de las diferencias observadas. Los factores analizados han sido el pez utilizado en primer lugar (Pez 1), los grados que ha sido rotada la imagen que contiene a ese pez (Grados 1) y la escala utilizada (Escala 1). Los resultados del análisis de la varianza están tabulados en la tabla 12.6.

Tabla 12.6: Tabla ANOVA de los resultados de la función basada en el ajuste de los ángulos al comparar arenque con distintas orientaciones y escalas.

Variable dependiente: $\varepsilon_{1_k}^2$					
Fuente	Suma de cuadrados tipo II	gl	Media cuadrática	F	Sig.
Pez 1 (P_1)	7.551E-03	1	7.551E-03	10.978	.001
Grados 1 (G_1)	2.377E-02	3	7.925E-03	11.522	.000
Escala 1 (E_1)	9.710E-03	2	4.855E-03	7.059	.001
$P_1 * G_1$	7.413E-03	3	2.471E-03	3.592	.014
$P_1 * E_1$	1.965E-02	2	9.824E-03	14.283	.000
$G_1 * E_1$	4.323E-02	6	7.205E-03	10.476	.000
$P_1 * G_1 * E_1$	2.736E-02	6	4.560E-03	6.630	.000
Error	.363	528	6.878E-04		

Los niveles de significancia obtenidos por el análisis de la varianza son extremadamente pequeños (Sig. < 0.05), y, por tanto, todos los factores y las interacciones entre ellos son significativos y hay una gran evidencia de que los niveles de los factores no tienen la misma media.

El análisis de la varianza indica claramente que la función es sensible a la orientación y al tamaño inicial del objeto. Sin embargo, los factores que influyen en esta sensibilidad son ajenos a la función de comparación y son:

- La naturaleza discreta de la imagen que depende de la definición espacial (número de puntos de la imagen) y de la resolución digital (número de niveles utilizados en cada plano de color).
- El algoritmo de detección de bordes en color.
- El algoritmo de umbralización.
- El algoritmo de extracción del contorno exterior.

Por tanto, para poder aplicar correctamente la función basada en el ajuste de los ángulos es necesario que los objetos a comparar tengan la misma orientación (por ejemplo, horizontal) y un tamaño similar.

Las tablas 12.7 y 12.8 muestran, respectivamente, los subconjuntos homogéneos de los factores Grado 1 y Escala 1 obtenidos mediante el test de Scheffé. Como se puede observar, para cada factor, hay dos subconjuntos, que, además, son pocos homogéneos, ya que los niveles de significancia distan mucho del valor 1.

Aunque la orientación y el tamaño iniciales de los objetos a comparar influyen en la función de comparación, los resultados obtenidos demuestran que existe una gran similitud entre las diferentes imágenes, ya que el valor medio más grande obtenido para las diferentes orientaciones o escalas es inferior a 0’06 (figuras 12.4 y 12.5).

Tabla 12.7: Subconjuntos homogéneos para los valores de Grados 1 obtenidos mediante el test de Scheffé: se muestran las medias de los grupos en los subconjuntos homogéneos.

Grados 1	Subconjunto	
	1	2
0	3.153E-02	4.101E-02
30	3.370E-02	
45	4.101E-02	
90		4.821E-02
Sig.	0.030	0.159

Tabla 12.8: Subconjuntos homogéneos para los valores de Escala 1 obtenidos mediante el test de Scheffé: se muestran las medias de los grupos en los subconjuntos homogéneos.

Escala 1	Subconjunto	
	1	2
0.5	3.347E-02	3.863E-02
1.0	3.863E-02	
1.5		
Sig.	0.170	0.175

12.3.3 Pruebas con imágenes reales

La tercera prueba ha consistido en la utilización de la metodología para comparar los contornos de peces obtenidos a partir de imágenes reales. Los peces seleccionados fueron 2 caballas, 2 doradas y 2 lubinas. Para cada uno de los peces, se

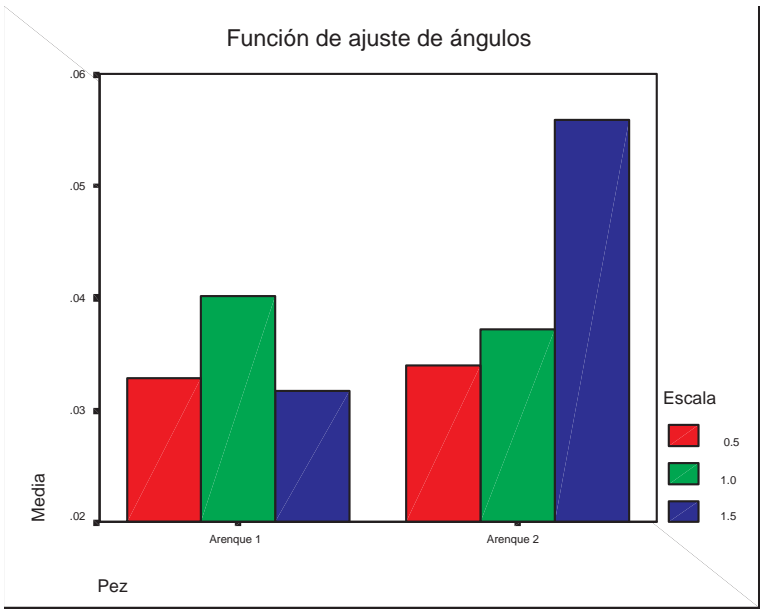


Figura 12.4: Media de los valores de la función basada en el ajuste de los ángulos obtenidos por cada pez para diferentes escalas.

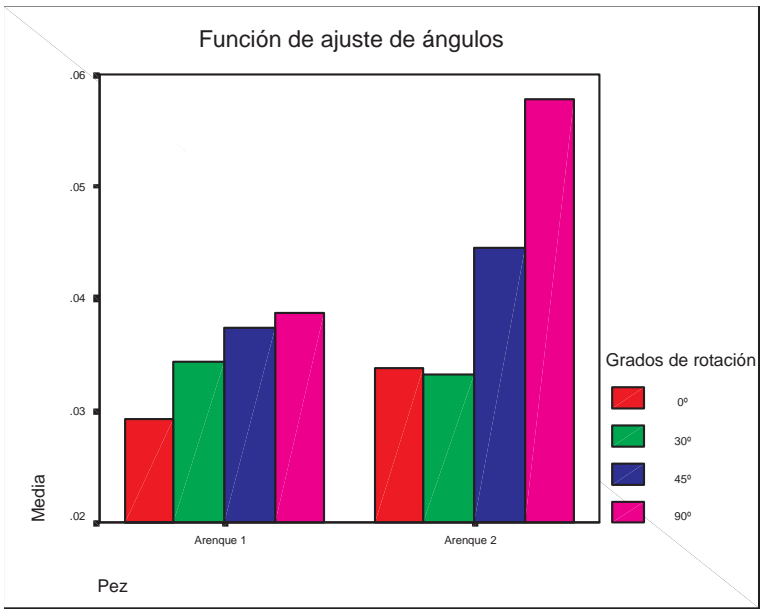


Figura 12.5: Media de los valores de la función basada en el ajuste de los ángulos obtenidos por cada pez para diferentes orientaciones.



(a)



(b)



(c)

Figura 12.6: Imágenes reales en color de peces utilizados en la tercera prueba: (a) caballa; (b) dorada; (c) lubina.

capturaron 3 imágenes en las que la orientación del pez era arbitraria (figura 12.6).

Las características de las imágenes reales fueron:

- Imágenes en color en el espacio *RGB* (formato *kdf de Khoros 2.2*).
- Número de niveles en cada plano de color: 256.
- Tipo de dato de los niveles de color: real de punto flotante (*float*).
- Altura: 768.
- Anchura: 572.

El contorno extraído de cada una de las 18 imágenes reales (figura 12.7) fue comparado con el de las demás utilizando las dos funciones de comparación diseñadas.

Los parámetros utilizados en las distintas fases del proceso de comparación de los contornos fueron los mismos que se indicaron en la primera prueba, salvo que el número máximo de iteraciones se fijó en 100.

El estudio de los resultados obtenidos al utilizar las dos funciones de comparación repiten las conclusiones que se indicaron en la primera prueba (subsección 12.3.1):

- Las dos funciones de comparación están estrechamente correlacionadas, como refleja el coeficiente de correlación de Pearson:

$$Pearson(\varepsilon_{1_k}^2, \varepsilon_{2_k}^2) = 0.999 \quad (12.7)$$

Esta correlación es significativa al nivel 0.01 (bilateral).

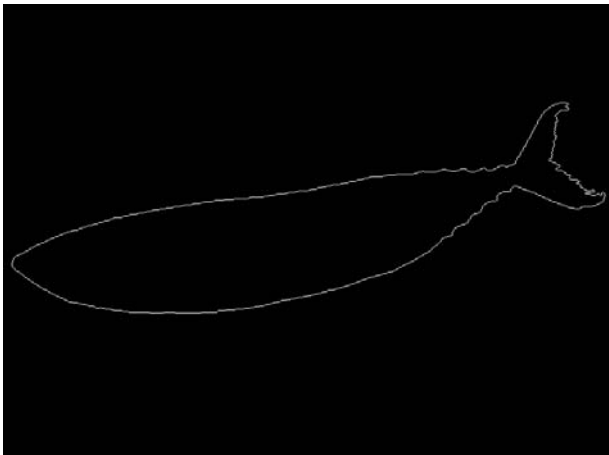
El valor de correlación 0.999 es próximo a 1 y, por tanto, las dos funciones de comparación obtienen resultados casi idénticos. Por tanto, conviene utilizar la función basada en el ajuste de los ángulos para comparar dos contornos entre sí, ya que su complejidad computacional es inferior.

- Para la función basada en el ajuste de los ángulos, los valores de los ángulos θ_{1_k} y θ_{2_k} que han de rotarse los contornos para alcanzar el valor mínimo también están correlacionados:

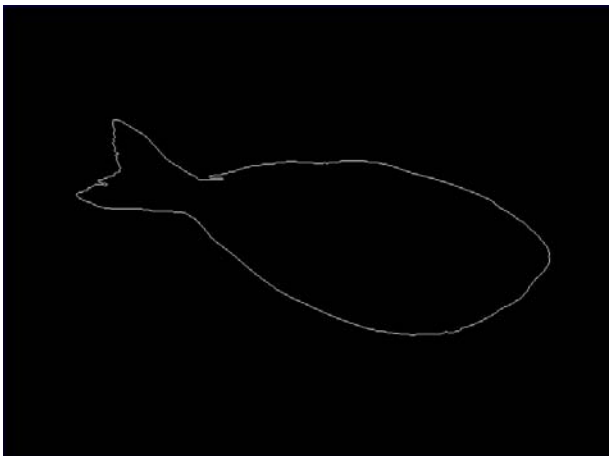
$$Pearson(\theta_{1_k}, \theta_{2_k}) = -1.000 \quad (12.8)$$

Esta correlación es significativa al nivel 0.01 (bilateral).

Al igual que ocurrió en la primera prueba, el valor -1.000 refleja que los ángulos son opuestos entre sí. Al normalizarse el tamaño de los contornos,



(a)



(b)



(c)

Figura 12.7: Contornos extraídos de las imágenes reales de la figura 12.6: (a) caballa; (b) dorada; (c) lubina.

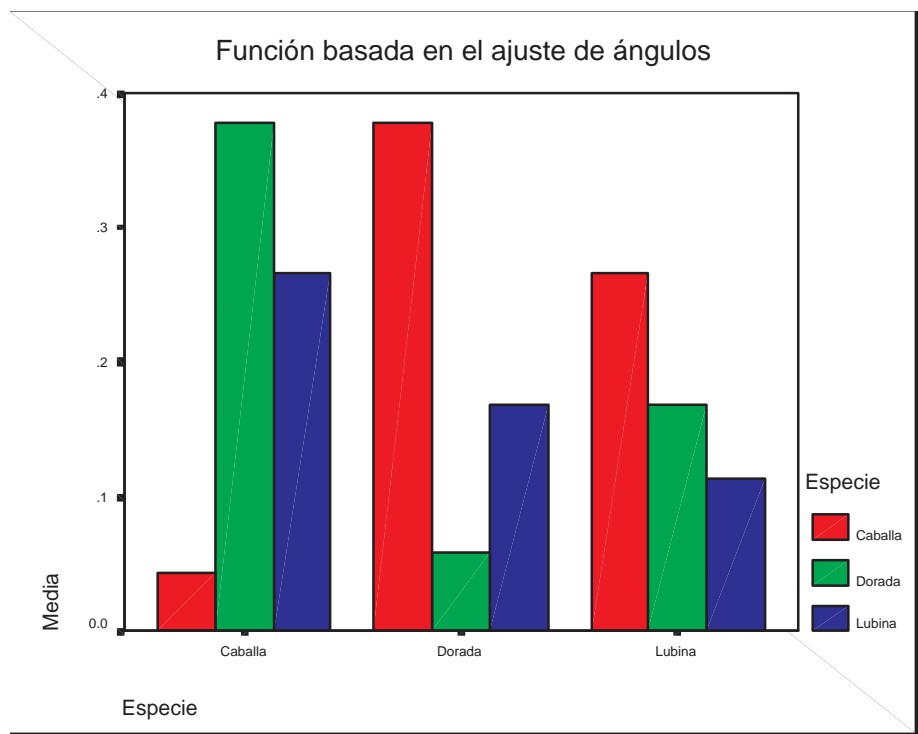


Figura 12.8: Media de los valores de la función de ajuste de los ángulos para cada una de las especies.

- Si se examina la media de los valores obtenidos de forma individual por cada uno de los dos peces de cada especie (figura 12.9 y tabla 12.10), se deduce que la función de comparación permite discriminar claramente las caballas y las doradas, presentando dificultades con las lubinas. Ello puede ser debido a similitudes entre las formas de los peces o a problemas en la captura de la imagen, como son sombras, brillos, etc. No obstante, se obtienen los resultados más pequeños cuando se utilizan imágenes correspondientes al mismo pez. Por ejemplo, los valores medios (tabla12.10) obtenidos al comparar diferentes contornos del pez *caballa 1* entre sí son 5'207 veces inferiores a los valores medios obtenidos al compararlos con los contornos del pez *caballa 2*, 28'049 veces con los contornos del pez *dorada2* o 21'758 veces con los contornos del pez *lubina 1*.

Para finalizar, se ha de indicar que, en un 73.9% de los casos, el número de iteraciones que se han necesitado para alcanzar el valor mínimo han sido 2 (tabla 12.11). Si se tiene en cuenta que la segunda iteración se utiliza para comprobar el criterio de parada del proceso de minimización, en realidad sólo se ha necesitado una única iteración para obtener el valor mínimo.

Tabla 12.10: Descriptores estadísticos de la función de ajuste de los ángulos para cada uno de los peces.

Pez 1	Pez 2	Media	Desviación típica	Mínimo	Máximo	Rango
Caballa 1	Caballa 1	.011916	.006366	.004842	.019076	.014234
	Caballa 2	.062048	.023232	.038713	.110846	.072133
	Dorada 1	.542439	.060744	.410683	.615688	.205005
	Dorada 2	.334243	.049345	.252776	.404285	.151509
	Lubina 1	.259279	.052018	.188854	.326211	.137357
	Lubina 2	.350237	.034241	.298501	.407655	.109154
Caballa 2	Caballa 1	.062048	.023232	.038713	.110846	.072133
	Caballa 2	.020058	.013908	.006767	.037158	.030391
	Dorada 1	.401079	.064444	.293125	.483508	.190383
	Dorada 2	.232518	.023596	.196843	.252763	.055920
	Lubina 1	.129854	.053700	.068548	.197527	.128979
	Lubina 2	.321360	.036649	.257477	.378345	.120868
Dorada 1	Caballa 1	.542439	.060744	.410683	.615688	.205005
	Caballa 2	.401079	.064444	.293125	.483508	.190383
	Dorada 1	.030379	.014562	.011620	.040824	.029204
	Dorada 2	.075003	.023701	.051267	.114717	.063450
	Lubina 1	.178525	.028602	.135856	.220403	.084547
	Lubina 2	.167638	.033153	.126987	.232151	.105164
Dorada 2	Caballa 1	.334243	.049345	.252776	.404285	.151509
	Caballa 2	.232518	.023596	.196843	.252763	.055920
	Dorada 1	.075003	.023701	.051267	.114717	.063450
	Dorada 2	.034603	.006721	.026303	.040945	.014642
	Lubina 1	.175928	.019959	.145147	.201592	.056445
	Lubina 2	.148349	.029339	.098593	.203556	.104963
Lubina 1	Caballa 1	.259279	.052018	.188854	.326211	.137357
	Caballa 2	.129854	.053700	.068548	.197527	.128979
	Dorada 1	.178525	.028602	.135856	.220403	.084547
	Dorada 2	.175928	.019959	.145147	.201592	.056445
	Lubina 1	.070636	.039828	.032477	.119561	.087084
	Lubina 2	.155099	.042129	.090483	.228377	.137894
Lubina 2	Caballa 1	.350237	.034241	.298501	.407655	.109154
	Caballa 2	.321360	.036649	.257477	.378345	.120868
	Dorada 1	.167638	.033153	.126987	.232151	.105164
	Dorada 2	.148349	.029339	.098593	.203556	.104963
	Lubina 1	.155099	.042129	.090483	.228377	.137894
	Lubina 2	.025412	.009628	.013953	.035313	.021360

Tabla 12.11: Frecuencias de las iteraciones necesitadas para el alcanzar el valor mínimo durante las pruebas con imágenes reales usando la función de ajuste de los ángulos.

Iteraciones	Frecuencia	Porcentaje
2.00	226	73.9
5.00	2	.7
14.00	2	.7
16.00	2	.7
17.00	2	.7
18.00	2	.7
33.00	2	.7
36.00	2	.7
40.00	2	.7
43.00	2	.7
44.00	4	1.3
47.00	2	.7
49.00	4	1.3
51.00	2	.7
54.00	2	.7
55.00	2	.7
57.00	2	.7
64.00	4	1.3
74.00	2	.7
99.00	38	12.4
Total	306	100.0

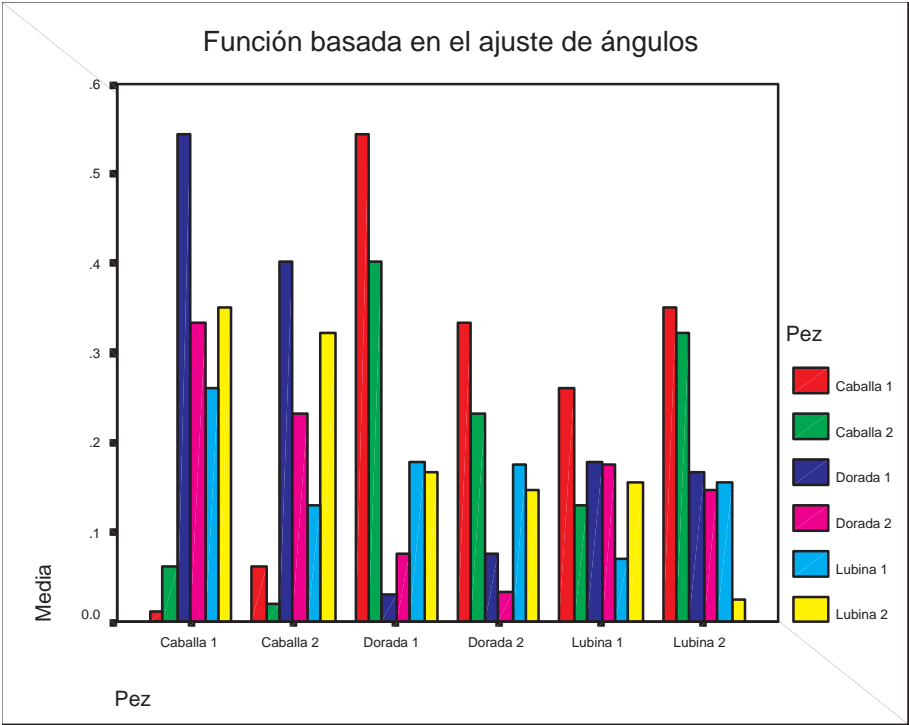


Figura 12.9: Media de los valores de la función de ajuste de los ángulos para cada uno de los peces.

12.4 Análisis de los resultados

Analizando los resultados de las tres pruebas realizadas, se pueden extraer las siguientes conclusiones:

- Para comparar dos contornos, sólo se ha de aplicar la función de comparación basada en el ajuste de los ángulos, ya que la función basada en el ajuste de los ángulos y los radios está correlacionada con la primera porque los radios de los puntos críticos han sido previamente normalizados.
- La función basada en el ajuste de los ángulos sólo necesita determinar el valor de θ_{1_k} , que representa la rotación que ha de efectuar el primer contorno para ajustarse al segundo. Ello es debido a que el valor de θ_{2_k} es el opuesto de θ_{1_k} .
- Además, al ser el valor obtenido por θ_{1_k} muy próximo a 0 y ser el número de iteraciones necesitadas en la mayoría de los casos igual a 2, se puede prescindir del proceso de minimización de la función de comparación si sólo se desea saber si dos contornos corresponden al mismo objeto.
- La orientación y el tamaño iniciales de los objetos influyen en los valores obtenidos por la función de comparación.

- La comparación de objetos contenidos en imágenes reales en color ha generado resultados aceptables. En algunos casos se han presentado dificultades provocadas por similitudes entre las formas de los peces o por problemas en la captura de las imágenes (sombras, brillos, etc.).

Parte IV

Conclusiones y futuros estudios

Capítulo 13

Conclusiones y futuros estudios

13.1 Conclusiones

Como consecuencia del trabajo desarrollado en esta Tesis Doctoral, se han obtenidos las siguientes conclusiones:

1. Se propone una extensión del método de Venkatesh y Kitchen (1.992) para evaluar algoritmos de detección de bordes. El nuevo método propuesto, a diferencia del método original, puede ser aplicado a bordes con cualquier forma geométrica.
2. En virtud de los resultados obtenidos, se puede concluir que la medida de evaluación propuesta por Baddeley (1.992a, 1.992b) es la más adecuada para comparar algoritmos de detección de bordes.
3. Entre los algoritmos de detección de bordes en color, los métodos propuestos por Di Zenzo (1.986) y Lee y Cok (1.991), que están basados en el concepto de gradiente multidimensional, han sido los que han obtenido un mejor rendimiento al ser aplicados a imágenes en color artificiales y reales.
4. Se proponen dos versiones diferentes basadas en el algoritmo de adelgazamiento de bordes de O’Gorman (1.990).
5. Se proponen dos métodos para identificar los puntos del contorno exterior de un objeto y otro para crear un contorno cerrado a partir de contornos abiertos.
6. Se propone un algoritmo de representación del contorno que es invariante a traslaciones y rotaciones. Frente a propuestas clásicas que utilizan como primer punto del contorno a aquél que tiene mayor curvatura o ha sido localizado antes al recorrer la imagen, la propuesta que se hace está basada en el uso del centroide y el eje de mínima inercia.

7. Se ha comprobado que la función de comparación propuesta por Gu y Tjahjadi (2.000) no es simétrica ni está normalizada.
8. Se propone un método de “comparación” de contornos que utiliza como puntos característicos o críticos a los puntos relevantes que se obtienen del análisis de la curvatura, normaliza los radiovectores de dichos puntos y minimiza una función de comparación mediante el método del gradiente.
9. Se han propuesto dos funciones de comparación de contornos simétricas y normalizadas. La primera utiliza el ajuste de los ángulos de los puntos críticos mientras que la segunda utiliza el ajuste de los radiovectores de los mismos. Los resultados experimentales permiten deducir que ambas están correlacionadas. Además, sólo es necesario determinar el ángulo de rotación de los puntos característicos de un contorno, ya que el ángulo de rotación de los puntos críticos del otro contorno será el opuesto.

13.2 Futuros estudios

Debido a la estructura modular de la metodología de reconocimiento de objetos desarrollada, cualquier mejora o avance que se produzca en cualquiera de sus fases puede ser incorporada de forma inmediata.

A continuación se van a indicar algunas de las líneas de investigación que se pueden abordar para mejorar la metodología de reconocimiento de objetos bidimensionales desarrollada:

- Análisis de los métodos de supresión o atenuación del ruido en imágenes en color: aunque en el capítulo 2 se han citado algunos métodos que permiten suavizar una imagen en color, sería deseable analizar en profundidad su rendimiento.
- Extracción del contorno exterior: el método de extracción descrito en el capítulo 10 ha sido diseñado para imágenes que contienen un único objeto. Un problema más complejo ha de ser resuelto cuando la imagen contiene más de un objeto.
- Obtención de puntos característicos: se ha utilizado el algoritmo de Mokhtarian y Mackworth (1.992) para extraer los puntos de curvatura extrema de un contorno (capítulo 11), pero éste no es el único método posible, como ya se indicó en el capítulo 6. Por tanto, sería conveniente realizar un estudio para determinar qué método permite extraer de forma más eficiente los puntos característicos de un contorno, así como determinar qué tipo de puntos caracterizan mejor al contorno.

- Funciones de comparación de contornos: las dos funciones propuestas en el capítulo 11 están basadas en los puntos característicos obtenidos en diferentes niveles del espacio de escalas gaussiano. Habría que analizar hasta qué nivel del espacio de escalas el rendimiento es bueno sin perder información relevante del contorno.

Por último, se ha de indicar que la metodología desarrollada sólo utiliza el contorno para reconocer un objeto. Una mejora considerable sería la utilización del contorno junto con otras características, como el color o la textura, para reconocer un objeto.

Apéndice A

Color

A.1 Definición y características

La “luz” es una “energía radiante” que provoca la sensación de la visión al incidir en la retina y estimular los nervios ópticos. La percepción humana del “color” está producida por la variación en el espectro electromagnético de la luz. La luz visible ocupa una banda de longitudes de onda (o de frecuencias) relativamente estrecha del espectro electromagnético (aproximadamente entre 400 nm y 700 nm) que está delimitada por los rayos ultravioleta y los rayos infrarrojos (Teuber 1992).

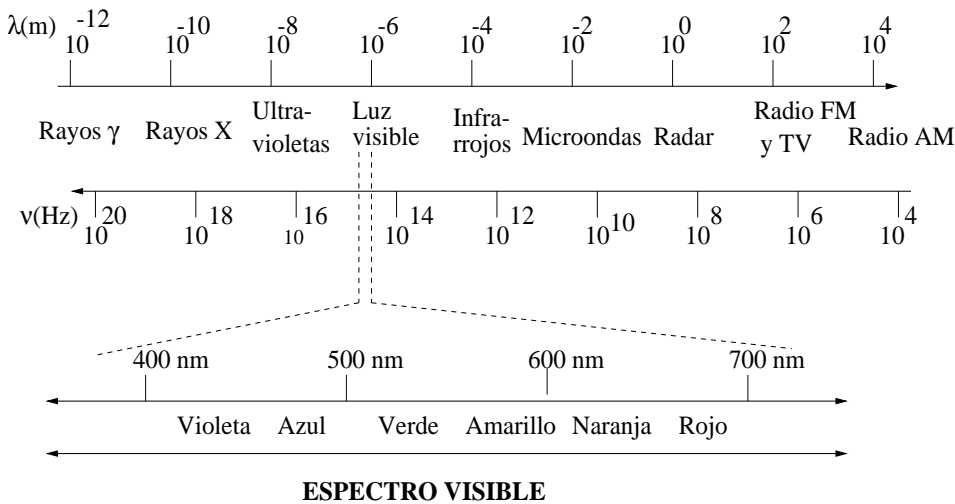


Figura A.1: Espectro electromagnético.

La principal característica que las personas pueden apreciar en el color es la facultad con que las dota para poder diferenciar dos objetos que posean tamaños y formas iguales.

En general, los colores que los seres humanos percibimos en un objeto están determinados por la naturaleza de la luz reflejada en el mismo. En este sentido, se dice que un objeto es verde cuando refleja la luz del espectro correspondiente a la banda en la que está situado el color verde, absorbiendo la luz que esté fuera de esa banda. Por consiguiente, el término color es algunas veces referido como “color percibido” para distinguirlo del color usado en el sentido de “color psicofísico” (Wyszecki y Stiles 1982).

La caracterización de la luz es un aspecto central de la ciencia del color. Si la luz es “acromática” (sin color), su único atributo es la “intensidad” (I) o cantidad de luz. El término “nivel de gris” se refiere a una medida escalar de la intensidad que va del negro al blanco, pasando por los grises.

Para describir las características de una fuente cromática se emplean tres magnitudes básicas (Wyszecki y Stiles 1982):

- Radiancia (*radiance*): cantidad total de energía que sale de una fuente luminosa.
- Luminancia (*luminance*): cantidad de energía que un observador percibe de una fuente luminosa. La luminancia suele ser denotada por Y .
- Brillo (*brightness*): se define como el atributo que representa la sensación visual según la cual un área parece emitir más o menos luz. Es un descriptor subjetivo de difícil medida y que incluye la noción acromática de la intensidad y es uno de los factores fundamentales para describir las sensaciones del color.

Aunque en principio puedan parecer similares, la radiancia y la luminancia son conceptos diferentes, como pone de manifiesto el siguiente ejemplo: una fuente luminosa puede emitir gran cantidad de energía (radiancia) en forma de rayos infrarrojos, pero dicha energía no puede ser percibida por un ser humano (luminancia) al estar fuera de los límites del espectro visible.

La representación del color está basada en la teoría de Thomas Young (1.802) que establece que cualquier color puede ser producido por la mezcla de “tres colores primarios” C_1 , C_2 y C_3 en sus proporciones adecuadas (Jain 1989):

$$C = aC_1 + bC_2 + cC_3 \quad (\text{A.1})$$

Las reglas esenciales de la colorimetría fueron establecidas por Grassman en 1.853 (Chapron 1992; Grassman 1853). Estas leyes son:

- cualquier color puede ser creado por otros tres colores y esta combinación de los tres colores es única,

- si dos colores son equivalentes, entonces también serán equivalentes después de multiplicar o dividir por un mismo número las tres componentes que los integran
- y la luminancia de una mezcla de tres colores es igual a la suma de las luminancias de cada color.

Esta teoría es consistente con el hecho de que el ojo humano tiene tres diferentes tipos de conos en la retina. Las clases de conos se diferencian en el tipo de proteína que compone el pigmento que capta la luz y en el rango de longitudes de onda a las que es sensible dicho pigmento. La tabla A.1 muestra la sensibilidad máxima de cada clase de cono.

Tabla A.1: Sensibilidades máximas de los tipos de conos de la retina.

Nombre (identificación)	Máxima sensibilidad	Alias
β (azul)	455 nm (violeta)	S (Short)
γ (verde)	535 nm (verde)	M (Medium)
ρ (rojo)	570 nm (amarillo)	L (Long)

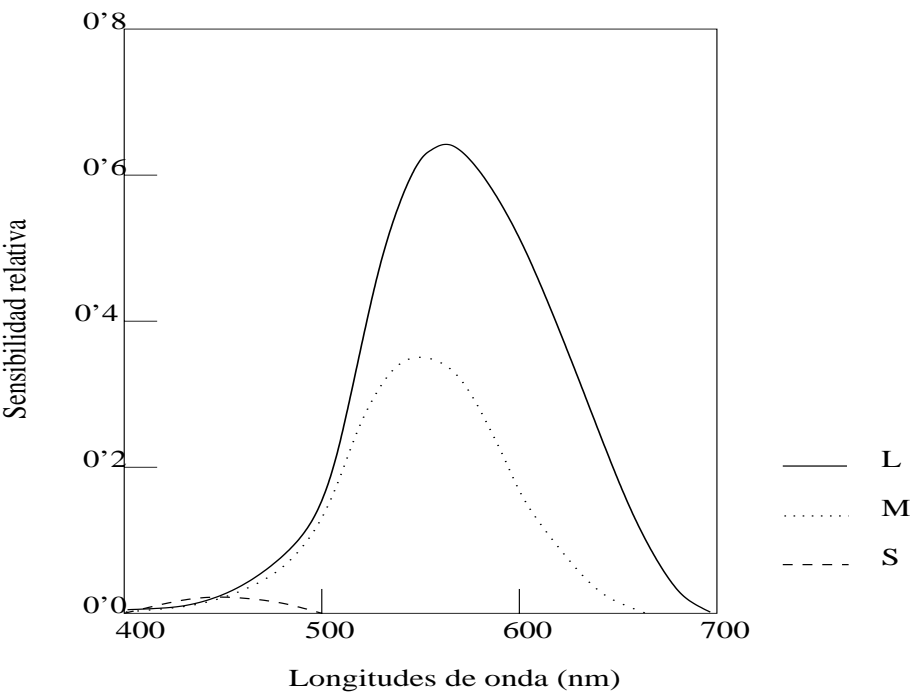


Figura A.2: Sensibilidad de los conos de la retina.

La cantidad de conos de cada tipo se ha estimado en la proporción:

$$\rho : \gamma : \beta \rightarrow 40 : 20 : 1 \tag{A.2}$$

Debido a la estructura del ojo humano, todos los colores se ven como combinaciones variables de los denominados “tres colores primarios”: “rojo”, “verde” y “azul”.

La Comisión Internacional de la Iluminación *C.I.E.* (*Commission Internationale de l'Eclairage*) propuso el uso normalizado del color y realizó un estudio en 1931 en el cual un conjunto de personas actuaban como “observadores” encargados de “identificar” los colores primarios. A partir de dicho estudio, *C.I.E.* designó las siguientes longitudes de ondas para los tres colores primarios (González y Woods 1996):

- rojo = 700 nm,
- verde = 546'1 nm
- azul = 435'8 nm

Posteriormente, en 1964, *C.I.E.* realizó un segundo estudio de colorimetría que provocó una renovación de los valores asignados a los colores primarios (Wyszecki y Stiles 1982):

- rojo = 645'2 nm,
- verde = 526'3 nm
- azul = 444'4 nm

Sin embargo, una longitud de onda específica no puede identificar de manera unívoca un color que pueda llamarse rojo, verde o azul, ya que cada uno de estos colores abarca una pequeña franja del espectro visible. Además, tres componentes de *RGB* “fijas” no pueden generar por sí solas todos los colores del espectro.

Los colores primarios se pueden “sumar” para obtener “los colores secundarios” de la luz: magenta (rojo más azul), cian (verde más azul) y amarillo (rojo más verde). Mezclando los tres colores primarios, o uno secundario con su color primario opuesto, en las proporciones adecuadas, se obtiene luz blanca.

A.2 Espacios de color

El procesamiento digital de imágenes en color requiere la utilización de “un espacio o modelo de color”. Un modelo de color define un sistema de coordenadas tridimensional y un subespacio de ese sistema en el que cada color queda representado por un único punto. El uso de un modelo de color permite medir y especificar cuantitativamente los colores de una forma normalizada.

Se han propuesto diferentes espacios de color, los cuales se pueden agrupar en dos categorías:

- Independientes del dispositivo:
 - XYZ y xyz .
 - $L^*u^*v^*$ y $L^*a^*b^*$.
 - Etc.
- Dependientes del dispositivo:
 - RGB
 - rgb , Irg e YT_1T_2 .
 - YIQ e YUV
 - HSI , HLS , HSV y $TekHVC$
 - CMY
 - Espacios de colores contrarios.
 - Etc.

Los modelos dependientes del dispositivo necesitan especificar un color directamente en el espacio de color de dicho dispositivo. En el caso de los dispositivos que emiten luz (monitores), se puede definir un espacio geométrico aditivo. En los dispositivos que reflejan la luz (impresoras), no se puede crear dicho espacio aditivo, sino que se ha de utilizar un espacio geométrico sustractivo, en el cual se restan componentes de la luz blanca.

Una característica importante que deben poseer los espacios de color es que sean “visualmente uniformes”, es decir, las distancias numéricas entre colores han de estar relacionadas con la capacidad humana para percibir diferencias de colores. Los espacios que poseen esta característica se denominan “espacios de color de escala uniforme” (*USC, Uniform Scale Chromaticity*).

Los formatos de color están basados en un grafo de color de tres dimensiones sobre un cubo, excepto *HSI*, *HLS*, *HSV* y *TekHVC* que usan un modelo cilíndrico o cónico.

Los espacios de color más utilizados en el procesamiento digital de imágenes son los modelos *RGB*, *YIQ* y *HSI*.

Son pocos los estudios realizados para determinar el espacio de color más apropiado para el procesamiento digital de imágenes:

- Se analizaron nueve componentes de información del color (R , G , B , Y , I , Q , H , S e I) para determinar qué atributo del color era el más útil

para la segmentación de una imagen en regiones mediante el análisis de los histogramas. Los resultados obtenidos indicaron que el atributo H (*hue*, tono) era el más útil para tal tipo de segmentación (Ohlander 1975).

- Ohta (1980,1985) realizó un estudio comparativo de los espacios de color como consecuencia del cual propuso un nuevo modelo de color que es descrito en la subsección A.2.8.
- Gauch y Hsia (1992) utilizaron tres algoritmos de segmentación para comparar cuatro espacios de color: HLS , Lab , RGB e YIQ . La principal conclusión de su trabajo fue que la selección del espacio de color dependía del tipo de imagen: si la imagen era real entonces los mejores resultados se conseguían utilizando los espacios de color HLS y Lab ; si la imagen era artificial, los mejores resultados se obtenían con los modelos de color YIQ y RGB . Los propios autores indicaron que esta importante conclusión debía ser contrastada mediante la utilización de un gran número de imágenes, ya que ellos sólo utilizaron una imagen real y otra artificial.

A.2.1 Espacio de color RGB

El espacio de color RGB debe su nombre a las iniciales de los tres colores primarios: *Red* (rojo), *Green* (verde) y *Blue* (azul). Este modelo está basado en el sistema tridimensional de coordenadas cartesianas y cada color aparece con sus componentes espectrales primarias de rojo, verde y azul. El subespacio de color se muestra en la figura A.3.

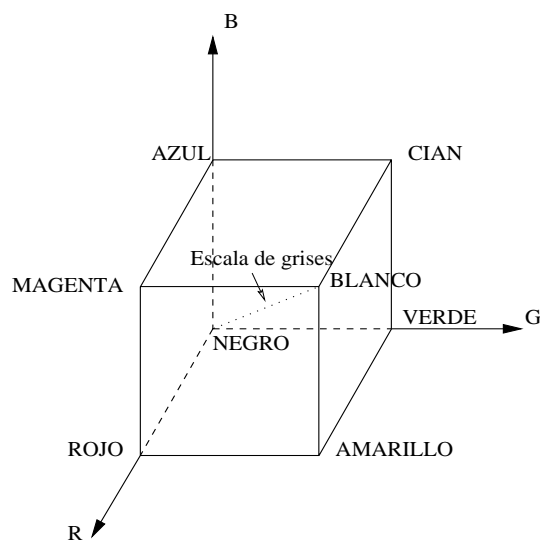


Figura A.3: Espacio de color RGB .

El espacio de color RGB es el modelo de color más popular, ya que es el formato estándar de los monitores en color y de la mayoría de las cámaras de vídeo

empleadas para la adquisición de imágenes digitales en color. Esto dos hechos influyen notablemente en su elección para el procesamiento digital de imágenes.

Las características principales de este espacio de color son:

- No requiere ninguna transformación para ser utilizado en el procesamiento digital de imágenes.
- Las componentes R , G y B poseen un fuerte factor de intensidad y están altamente correlacionadas (Ohta 1985).
- No es perceptualmente uniforme.

A.2.2 Espacios de color rgb , Irg e YT_1T_2

El espacio de color rgb también se denomina espacio de color “ RGB normalizado” debido a su definición:

$$r = \frac{R}{R + G + B} \quad (\text{A.3})$$

$$g = \frac{G}{R + G + B} \quad (\text{A.4})$$

$$b = \frac{B}{R + G + B} \quad (\text{A.5})$$

Las características de este espacio de color son:

- No es sensible a cambios de intensidad, puesto que sus componentes r , g y b sólo dependen de la proporción de las componentes R , G y B .
- Es inestable y poco significativo cuanto la intensidad es pequeña.

Una de las componentes del espacio rgb es redundante, ya que

$$r + g + b = 1 \quad (\text{A.6})$$

Debido a ello, se suele incorporar la intensidad, obteniéndose el espacio de color Irg , donde

$$I = \frac{R + G + B}{3} \quad (\text{A.7})$$

Una generalización del espacio de color Irg es el espacio de color YT_1T_2 (Nevatia 1977):

$$Y = c_1R + c_2G + c_3B \quad (\text{A.8})$$

$$T_1 = \frac{R}{R + G + B} \quad (\text{A.9})$$

$$T_2 = \frac{G}{R + G + B} \quad (\text{A.10})$$

donde

$$c_1 + c_2 + c_3 = 1 \quad (\text{A.11})$$

En ambos espacios de color, la primera componente (I o Y) representa la intensidad o luminancia, mientras que las otras dos (r o T_1 y g o T_2) representan la información cromática.

A.2.3 Espacios de color XYZ y xyz

Los espacios de colores RGB y rgb están basados en el uso de tres colores primarios, pero no existe ningún conjunto de tres colores que sea capaz de describir todos los colores que pueden ser percibidos. Debido a ello, *C.I.E.* propuso en 1.931 un conjunto estándar de colores primarios. Los colores primarios elegidos no son reales sino imaginarios, ya que, al estar demasiado saturados, no pueden ser vistos por el ojo humano o producidos (son físicamente irrealizables). Sin embargo, estos colores primarios imaginarios pueden describir matemáticamente todos los colores que pueden ser percibidos. Los colores primarios seleccionados fueron denotados por X , Y y Z y permiten definir el sistema de color XYZ .

Las características de este modelo de color son (Plataniotis y Venetsanopoulos 2000; Wyszecki y Stiles 1982):

- Ventajas:
 - Ha sido rigurosamente definido y es un estándar internacional.
 - Es el principal espacio para realizar medidas de color.
 - La información de la luminancia es determinada sólo por la componente Y de las otras dos componentes.
 - A partir de él se deduce “el diagrama cromático xy ”.
 - Es la base para otros espacios de *C.I.E.*
- Desventajas:
 - No es fácil especificar de manera intuitiva un nuevo color.

El sistema de coordenadas XYZ se obtiene a partir de una transformación lineal del sistema RGB . Sin embargo, la matriz de transformación debe ser

determinada empíricamente (Skarbek y Koschan 1994). La ecuación A.12 muestra la transformación¹ propuesta por *C.I.E.* (Pitas 1993; Wyszecki y Stiles 1982), mientras que la ecuación A.13 se corresponde con la transformación propuesta por el sistema NTSC (*National Television System Committee*) de televisión (Pitas 1993).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0'490 & 0'310 & 0'200 \\ 0'177 & 0'813 & 0'011 \\ 0'000 & 0'010 & 0'990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.12})$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0'607 & 0'174 & 0'201 \\ 0'299 & 0'587 & 0'114 \\ 0'000 & 0'066 & 1'116 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.13})$$

El color blanco es representado por $X = Y = Z = 1$.

Si sólo interesa la proporción de los colores primarios del conjunto estándar, entonces se pueden utilizar “los coeficientes tricromáticos”:

$$x = \frac{X}{X + Y + Z} \quad (\text{A.14})$$

$$y = \frac{Y}{X + Y + Z} \quad (\text{A.15})$$

$$z = \frac{Z}{X + Y + Z} \quad (\text{A.16})$$

resultando que

$$x + y + z = 1 \quad (\text{A.17})$$

El espacio de color xyz es inestable cuando la intensidad es pequeña.

Los coeficientes x e y se utilizan para obtener el diagrama cromático que es mostrado en la figura A.4. Las elipses de la figura A.4, también llamadas “elipses de MacAdam”, corresponden a colores que, aunque pueden ser vistos, no pueden ser distinguidos por el ojo humano (Jain 1989). Se puede observar que el tamaño y la orientación de las elipses es variable. Debido a este problema, las diferencias de colores no pueden ser definidas de una manera uniforme en el plano (x, y) (Pitas 1993).

A.2.4 Espacios de color $L^*u^*v^*$ y $L^*a^*b^*$

C.I.E. ha utilizado el sistema XYZ para definir espacios de color de escala uniforme (*USC*) que sí pueden medir las diferencias entre los colores de una forma continua: $L^*u^*v^*$ y $L^*a^*b^*$ (Wyszecki y Stiles 1982).

¹Se han propuesto otras matrices de transformación (Ohta et al. 1980; Plataniotis y Venet-sanopoulos 2000).

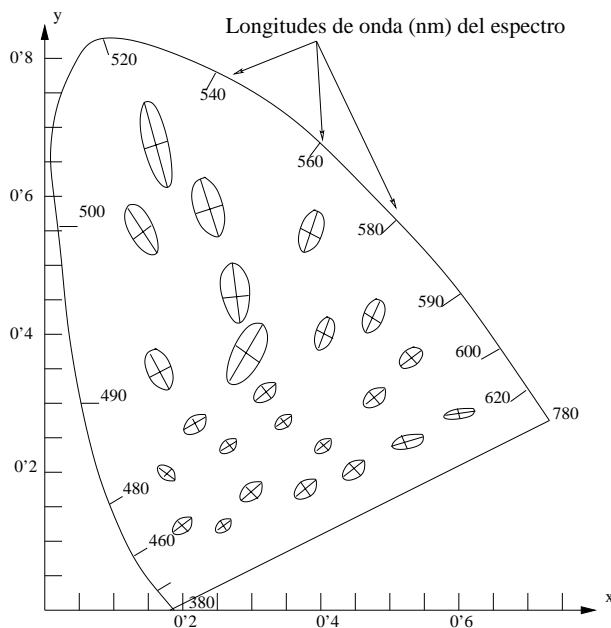


Figura A.4: Diagrama cromático del sistema XYZ.

Las conversiones del modelo XYZ a los modelos $L^*a^*b^*$ o $L^*u^*v^*$ no son complejas pero sí son lentas, porque requieren realizar operaciones con números reales de punto flotante. Además, los resultados obtenidos también son números reales de punto flotante. Este hecho provoca que las necesidades de memoria para las imágenes en color transformadas sean cuatro veces más grandes que las requeridas por la imagen en color original (Pitas 1993).

El sistema $L^*u^*v^*$, donde L^* representa la luminosidad y u^* y v^* , las coordenadas de cromaticidad, se define como:

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} - 16 & \text{si } \frac{Y}{Y_0} > 0'008856 \\ 903'3 \frac{Y}{Y_0} & \text{si } \frac{Y}{Y_0} \leq 0'008856 \end{cases} \quad (\text{A.18})$$

$$u^* = 13L^*(u' - u'_n) \quad (\text{A.19})$$

$$v^* = 13L^*(v' - v'_n) \quad (\text{A.20})$$

donde

$$u' = \frac{4X}{X + 15Y + 3Z} \quad v' = \frac{9X}{X + 15Y + 3Z} \quad (\text{A.21})$$

$$u'_n = \frac{4X_n}{X_n + 15Y_n + 3Z_n} \quad v'_n = \frac{9X_n}{X_n + 15Y_n + 3Z_n} \quad (\text{A.22})$$

Los valores X_n , Y_n y Z_n son los triestímulos correspondientes al color blanco.

El modelo $L^*u^*v^*$ es perceptualmente uniforme y permite utilizar la distancia euclídea para obtener las diferencias entre los colores:

$$(\Delta C)^2 = (\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2 \quad (\text{A.23})$$

El sistema $L^*a^*b^*$ también es usado para medir diferencias entre colores. L^* denota la luminosidad, mientras que a^* y b^* describen los contenidos rojo-verde y amarillo-azul de la luz, respectivamente. El valor de L^* se obtiene utilizando la ecuación A.18, mientras que las otras dos componentes se definen como

$$a^* = 500 \left[\left(\frac{X}{X_n} \right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} \right] \quad (\text{A.24})$$

$$b^* = 200 \left[\left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n} \right)^{\frac{1}{3}} \right] \quad (\text{A.25})$$

con

$$\frac{X}{X_n}, \frac{Y}{Y_n}, \frac{Z}{Z_n} > 0.01 \quad (\text{A.26})$$

donde X_n , Y_n y Z_n son los estímulos correspondientes al color blanco.

De nuevo se utiliza la distancia euclídea para calcular la diferencias entre colores:

$$(\Delta C)^2 = (\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2 \quad (\text{A.27})$$

El inconveniente de estos dos modelos es que las componentes u^* , v^* , a^* y b^* son inestables cuando la intensidad es pequeña.

A.2.5 Espacios de color YIQ e YUV

El significado de YIQ es:

- Y : luminancia (*luminance*).
- I : en - fase (*in - phase*).
- Q : cuadratura (*quadrature*).

El modelo YIQ se utiliza en las emisiones comerciales de televisión (sistema NTSC), ya que permite mantener la compatibilidad con las televisiones en blanco y negro. En realidad, la componente Y del sistema YIQ proporciona toda la información de vídeo que necesita un sistema de televisión monocromático.

El modelo YIQ fue diseñado para aprovechar la mayor sensibilidad del ojo humano a los cambios de iluminación frente a los cambios de tono (*hue*) o saturación. La principal ventaja del sistema YIQ en el procesamiento de imágenes en color es que la luminancia (Y) y la información del color (I y Q) están desacopladas. Esta característica permite que se pueda procesar la información de intensidad (Y) sin que ello afecte a los colores relativos de la imagen.

El espacio de color YIQ se obtiene a partir de una transformación lineal del modelo RGB (González y Woods 1996):

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0'299 & 0'587 & 0'114 \\ 0'596 & -0'275 & -0'321 \\ 0'212 & -0'523 & 0'311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.28})$$

A partir de las componentes I y Q se pueden obtener los valores de tono (H) y saturación (S):

$$H = \arctan \frac{Q}{I} \quad (\text{A.29})$$

$$S = \sqrt{I^2 + Q^2} \quad (\text{A.30})$$

Un modelo similar al espacio de color YIQ es el espacio de color YUV , que es el formato de color utilizado por el sistema de televisión PAL.

El significado de YUV es:

- Y : luminancia (*luminance*).
- U : tono (*hue*).
- V : saturación (*saturation*).

Al igual que YIQ , el modelo YUV tiene separada la información de luminancia (Y) de la información cromática (U y V).

La ecuación A.31 muestra la conversión de RGB a YUV (IC-PCI 1995):

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0'299 & 0'587 & 0'114 \\ -0'169 & -0'331 & -0'500 \\ 0'500 & 0'419 & 0'081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.31})$$

A.2.6 Espacio de color CMY

Este modelo toma su nombre de las iniciales de *Cyan* (cian), *Magenta* (magenta) y *Yellow* (amarillo), los cuales son considerados “los colores secundarios de la

luz” o “colores primarios de los pigmentos”. Si una luz blanca incide sobre una superficie pigmentada con cian, entonces la luz roja no es reflejada, es decir, cian “subtrae” la luz roja de la luz blanca. Un comportamiento similar se puede observar con el magenta y el verde o el amarillo y el azul.

La mayoría de los dispositivos de impresión (impresoras, fotocopadoras, *plotters*) requieren que los datos de entrada estén en formato *CMY* o realizan una conversión de *RGB* a *CMY* utilizando la ecuación A.32, donde se asume que los valores de los colores están normalizados en el rango $[0, 1]$.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.32})$$

A.2.7 Espacios de color *HSI*, *HLS*, *HSV* y *TekHVC*

Todos los modelos de color descritos anteriormente están influenciados por aspectos del dispositivo (*RGB*, *rgb*, *Irg*, YT_1T_2 , *YIQ*, *YUV*, *CMY*) o por razones de colorimetría (*XYZ*, $L^*u^*v^*$, $L^*a^*b^*$). Tales sistemas no describen la percepción humana del color y por ello se han diseñado otros espacios de color que sí lo permiten. Estos nuevos espacios de color utilizan los siguientes atributos:

- Tono: atributo relacionado con la longitud de onda dominante en una mezcla de ondas luminosas, es decir, representa el color dominante tal como lo percibe el observador. Cuando se dice que un objeto es rojo, verde o amarillo, se está indicando su tono,
- Saturación: indica la pureza relativa o cantidad de luz blanca mezclada con un tono. Los colores puros del espectro están completamente saturados, mientras que colores como el rosa (rojo y blanco) y el lavanda (violeta y blanco) están menos saturados, siendo su grado de saturación inversamente proporcional a la cantidad de luz blanca añadida.
- Brillo o luminosidad: representa la intensidad de luz percibida por el observador.

La cromaticidad está constituida por el tono y la saturación y, por tanto, un color se puede caracterizar por su brillo y su cromaticidad.

Los espacios de color que utilizan estos atributos son:

- *HSI*: tono (*hue*), saturación (*saturation*) e intensidad (*intensity*).
- *HLS*: tono (*hue*), luminosidad (*lightness*) y saturación (*saturation*).

- *HSV*: tono (*hue*), saturación (*saturation*) y valor (*value*).
- *TekHVC*: tono (*hue*), valor (*value*) y croma² (*chroma*) .

El sistema *HSI* es un sistema cilíndrico que se obtiene a partir del espacio de color *RGB* mediante la conversión del sistema de coordenadas cartesianas a un sistema de coordenadas cilíndricas. Las siguientes fórmulas muestran la conversión de *RGB* a *HSI* (Ohta 1985)³:

$$H = \arctan2(\sqrt{3}(G - B), 2R - G - B) \quad (\text{A.33})$$

$$S = 1 - \frac{\text{mínimo}(R, G, B)}{I} \quad (\text{A.34})$$

$$I = \frac{R + G + B}{3} \quad (\text{A.35})$$

Los colores visibles del sistema de coordenadas cilíndrico *HSI* son aquéllos que están incluidos en el cubo *RGB* (figura A.5).

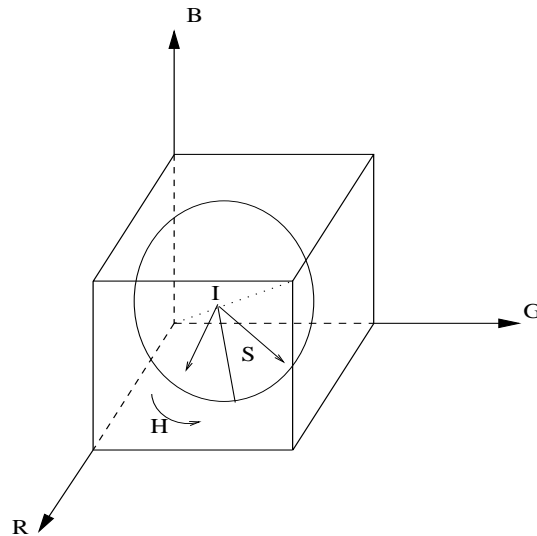


Figura A.5: Espacio de color *HSI*.

Los valores del tono (*hue*) varían entre 0 y 360 grados. El valor 0° corresponde al rojo. El rango de valores de la saturación es el intervalo [0, 1].

²Croma y saturación poseen significados muy similares (Wyszecki y Stiles 1982):

- Saturación: es un atributo de la sensación visual que permite indicar el grado de diferencia entre un estímulo cromático y un estímulo acromático sin tener en cuenta el brillo.
- Croma: es un atributo de la sensación visual que permite indicar la diferencia entre un estímulo cromático y un estímulo acromático del mismo brillo.

³Existen variantes de estas transformaciones, pero todas siguen la misma filosofía (Pitas 1993; Skarbek y Koschan 1994).

El sistema *HLS* es definido en una doble pirámide hexagonal, mientras que el sistema *HSV* lo es en una pirámide hexagonal invertida. La figura A.6 muestra los espacios de color *HLS* y *HSV*.

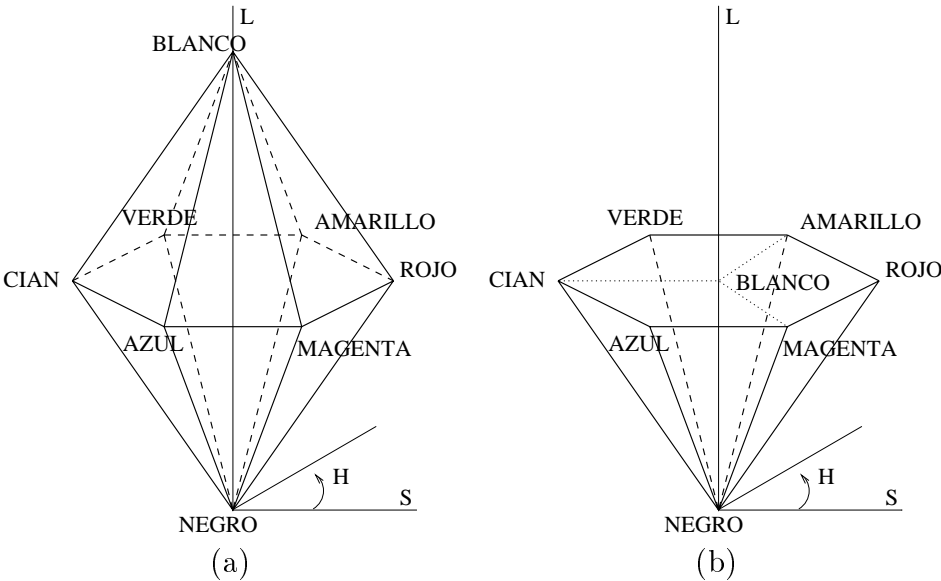


Figura A.6: Espacios de color: (a) *HLS* y (b) *HSV*.

Un modelo menos conocido que los anteriores es el espacio de color *TekHVC*, el cual ha sido desarrollado por Tektronik a partir del modelo *C.I.E. L*u*v** (Foley et al. 1996). La figura A.7 muestra este espacio de color.

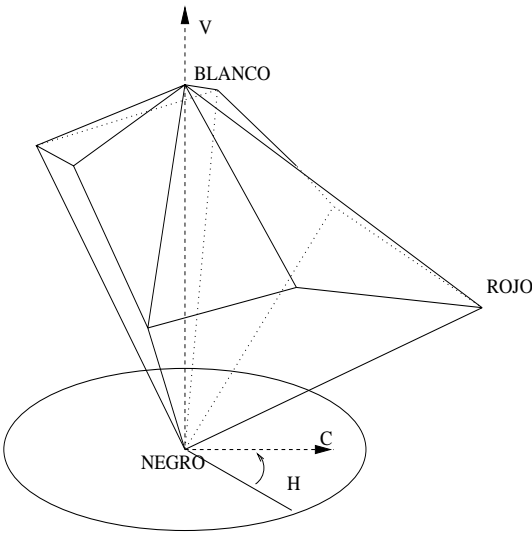


Figura A.7: Espacio de color *TekHVC*.

Los modelos *HSI*, *HLS*, *HSV* y *TekHVC* deben su utilidad a tres hechos básicos:

- La componente de la intensidad (I , L o V) está desacoplada de la información cromática contenida en la imagen.
- Las componentes de tono y saturación están íntimamente relacionadas con la forma en que las personas perciben el color.
- Son espacios de escala uniforme (USC).

Estos espacios de color están idealmente indicados para la mejora de imágenes en color, dado que la componente de intensidad (I , L o V) está desacoplada de la información cromática de la imagen (H y S). Se pueden aplicar las técnicas monocromáticas al plano de intensidad para mejorar imágenes en color real. El proceso sería el siguiente: convertir la imagen de RGB a HSI (HLS , HSV o $TeckHVC$), aplicar las técnicas de mejora al plano de intensidad y después deshacer el cambio. El contenido del color de la imagen no resulta afectado. Si se aplicarán directamente las técnicas de realce monocromáticas a los planos RGB , entonces los colores podrían quedar transformados al combinarse entre ellos, dando una apariencia de color poco natural (por ejemplo, alteración del color de la piel de las personas).

No obstante lo anterior, estos espacios de color presentan una singularidad en la componente de tono (*hue*) cuando los valores de RGB están próximos a cero (Skarbek y Koschan 1994). Se puede comprobar en la ecuación A.33 que el valor de H está indefinido cuando $R = G = B = 0$. Además, si se produce una pequeña variación del punto $(0, 0, 0)$ al punto $((1 - \alpha)\epsilon, \alpha\epsilon, 0)$, entonces el valor obtenido para la componente H es

$$H = \arctan2(\sqrt{3}\alpha, 2 - 3\alpha) \quad (\text{A.36})$$

Esto implica que para valores pequeños de ϵ el valor de tono puede variar desde cero (para $\alpha = 0$) a $2\pi/3$ (para $\alpha = 1$).

Otro inconveniente de estos espacios de color radica en el hecho de que las imágenes se han de convertir al formato RGB (o CMY) para que puedan ser visualizadas.

A.2.8 Espacios de colores contrarios

La familia de espacios de colores contrarios (*opponent color space*) está inspirada en la fisiología del sistema visual humano (Platanotis y Venetsanopoulos 2000). Según la teoría de los sistemas de visión en color (Hering 1977), el sistema visual humano puede ser expresado en términos de tonos de color contrarios, amarillo y azul por un lado y verde y rojo por otro, los cuales se anulan mutuamente cuando se superponen. Uno de los modelos de color basados en esta teoría consta de una

componente acromática (I) y de dos componentes de colores contrarios (RG , YB) definidas por (Holla 1982; Plataniotis y Venetsanopoulos 2000):

$$I = R + G + B \quad (\text{A.37})$$

$$RG = R - G \quad (\text{A.38})$$

$$YB = 2B - R - G \quad (\text{A.39})$$

Otro modelo de este tipo de espacio de color fue propuesto por Ohta (1980, 1985), el cual realizó un interesante estudio, basado en la transformación dinámica de Karhunen-Loeve, para determinar qué sistema de color era el más apropiado para la segmentación digital de imágenes. El resultado de su trabajo fue la aparición de un nuevo espacio de color derivado del modelo RGB :

$$I_1 = \frac{R + G + B}{3} \quad (\text{A.40})$$

$$I'_2 = R - B \quad (\text{A.41})$$

$$I'_3 = \frac{2G - R - B}{2} \quad (\text{A.42})$$

Ohta afirma que la característica I_1 es la más efectiva para la segmentación e I'_3 la menos efectiva.

Este modelo tiene como principal inconveniente que depende de la geometría de la escena (Healey 1992).

A.3 Imagen digital en el espacio de color RGB

Para poder procesar todas las longitudes de onda del espectro visible, éste se divide en varias bandas espectrales. Al capturar una imagen, se utilizan diferentes tipos de sensores, cada uno de los cuales es sensible a una determinada banda de longitudes de onda. De esta forma, cada banda se digitaliza independientemente de las demás y se representa del mismo modo que se hace con una imagen monocromática.

A la hora de representar la imagen en color en un dispositivo de visualización, se aplica un determinado método de combinación entre las diferentes imágenes o planos correspondientes a cada banda, dando como resultado la imagen en color.

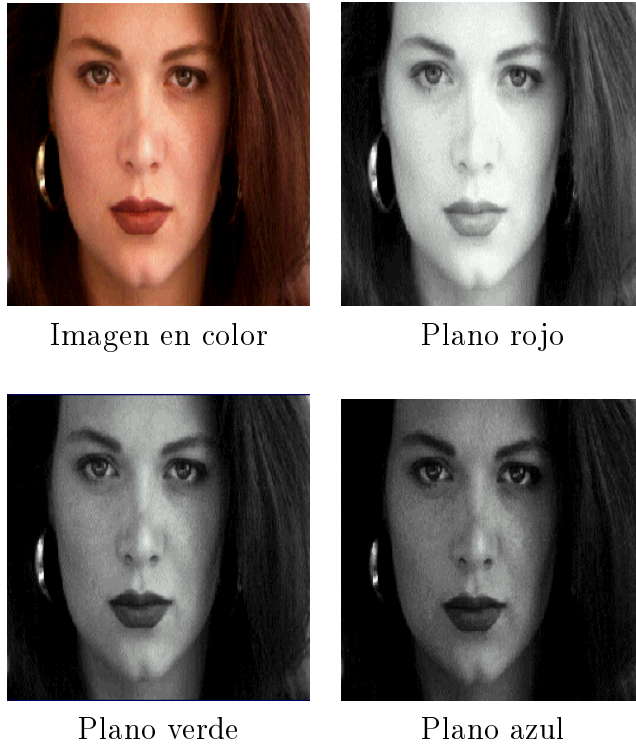


Figura A.8: Imagen en color y sus planos de color.

De este modo, se puede definir una imagen en color o multispectral como una función vectorial con tantas componentes como planos tenga dicha imagen. Si la imagen tiene n planos, se define una imagen en color $\vec{f}(x, y)$ como:

$$\vec{f}(x, y) = (f_1(x, y), f_2(x, y), \dots, f_n(x, y)) \quad (\text{A.43})$$

Las imágenes del modelo de color *RGB* constan de tres planos de color independientes entre sí, uno para cada color primario. Cada punto de una imagen en formato *RGB* está representado por una combinación de los colores rojo, verde y azul. La figura A.8 muestra una imagen en color y sus planos rojo, verde y azul.

Una imagen digital en color se puede representar por una matriz tridimensional en la que la primera dimensión indica uno de los tres planos (rojo, verde o azul), la segunda dimensión indica la fila y la tercera dimensión indica la columna. Si consideramos una imagen en color de $M \times N$ puntos, su representación podría indicarse de la siguiente forma:

$$\text{imagen_color}[0 \dots 2][0 \dots M - 1][0 \dots N - 1]$$

Como puede observarse por su definición, la primera dimensión de una imagen en color posee tres elementos numerados del 0 al 2. Cada uno de estos elementos representa uno de los tres planos de la imagen en color. La numeración de los planos es la siguiente:

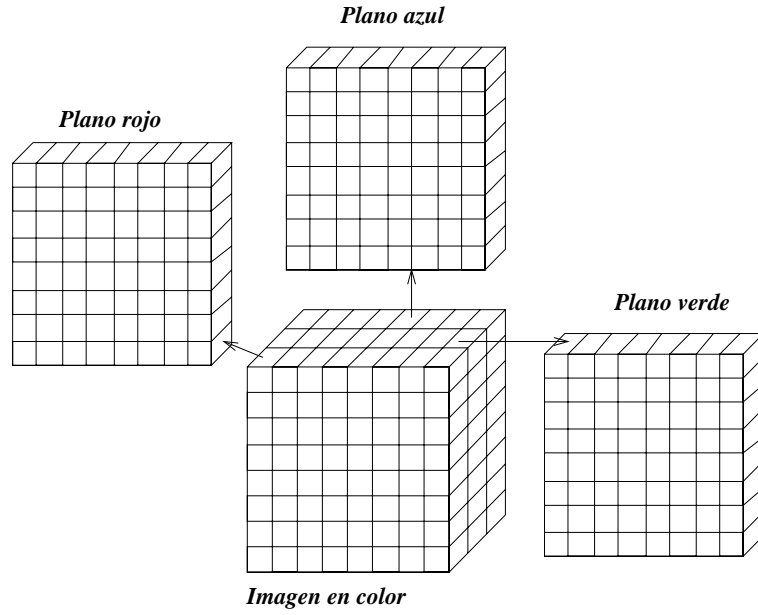


Figura A.9: Estructura de una imagen en color.

- 0: plano rojo
- 1: plano verde
- 2: plano azul

Por lo tanto, una imagen en color puede ser descompuesta en tres planos (ver figura A.9), siendo cada uno de ellos una matriz bidimensional de $M \times N$ elementos en la que la primera dimensión indica la fila de la matriz y la segunda la columna:

$$\begin{aligned} & \text{plano_rojo}[0 \dots M - 1][0 \dots N - 1] \\ & \text{plano_verde}[0 \dots M - 1][0 \dots N - 1] \\ & \text{plano_azul}[0 \dots M - 1][0 \dots N - 1] \end{aligned}$$

Llegando a un nivel de detalle mayor, una imagen en color puede considerarse como la unión de sus puntos. En una imagen en color, un punto p puede definirse como una estructura formada por tres campos que identifican el nivel de intensidad en cada una de los tres planos de la imagen, además de dos campos que representan las coordenadas espaciales del punto (ver figura A.10):

$$p \left\{ \begin{array}{l} R \text{ (Nivel de intensidad en el plano rojo)} \\ G \text{ (Nivel de intensidad en el plano verde)} \\ B \text{ (Nivel de intensidad en el plano azul)} \\ x \text{ (Número de fila)} \\ y \text{ (Número de columna)} \end{array} \right.$$

Los niveles de intensidad de un punto (x, y) de una imagen en color vienen dados por:

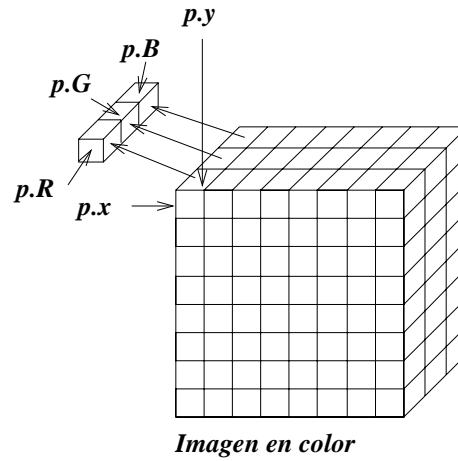


Figura A.10: Punto o *pixel* de una imagen en color.

$$\begin{aligned}
 p.R &= \text{imagen_color}[0][x][y] = \text{plano_rojo}[x][y] \\
 p.G &= \text{imagen_color}[1][x][y] = \text{plano_verde}[x][y] \\
 p.B &= \text{imagen_color}[2][x][y] = \text{plano_azul}[x][y]
 \end{aligned}$$

Los valores que pueden tomar los niveles de intensidad de cada punto de la imagen suelen pertenecer al intervalo $[0, 255]$.

Bibliografía

- Abdou, I. E. 1978. “Quantitative methods of edge detection”. Technical Report 830, Image Processing Institute, University of Souththerrn California.
- Abdou, I. E. y Pratt, W. K. 1979. “Quantitative design and evaluation of enhancement /thresholding edge detector”. *Proceedings of the IEEE*, 67(5):753–763.
- Abutaleb, A. S. 1989. “Automatic thresholding of gray-level pictures using two-dimensional entropy”. *Computer Vision, Graphics and Image Processing*, 47:22–32.
- Aho, A. V., Sethi, R. y Ullman, J. 1990. “*Compiladores: principios, técnicas y herramientas*”. Addison Wesley Iberoamericana.
- Allen, J. T. y Huntsberger, T. 1989. “Comparing color edge detection and segmentation methods”. En *IEEE Proceedings 1989 Souhteastcon*, pág. 722–727.
- Aloimonos, Y. 1988. “Visual shape computations”. *Proceedings of the IEEE*, 76:899–916.
- Alshatti, W. y Lambert, P. 1993. “Using eigenvectors of a vector field for deriving a second directional derivative operator for color images”. En Chetverikov, D. y Kropatsch, W. G., editores, *Proceedings of the 5th International Conference on Computer Analysis of Images and Patterns CAIP’93*, pág. 149–156, Budapest, Hungría.
- Arcelli, C. y Ramella, G. 1993. “Finding contour-based abstractions of planar pattern”. *Pattern Recognition*, 26(10):1563–1577.
- Asada, H. y Brady, M. 1986. “The curvature primal sketch”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:2–14.
- Astola, J., Haavisto, P. y Neuvo, Y. 1988. “Median type filters for color signals”. En *Proceedings of the 1988 IEEE Symposium on Circuits and Systems*, vol. 2(3), pág. 1753–1756.
- Astola, J., Haavisto, P. y Neuvo, Y. 1990. “Vector median filters”. En *Proceedings os the IEEE*, vol. 78, pág. 678–689.

- Atae-allah, C., Gómez-Lopera, J.-F., Robles-Pérez, A. y Román-Roldán, R. 1997. "Entropic edge detection and linking". En *VII National Symposium on Pattern Recognition and Image Analysis*, vol. II, pág. 16–17.
- Atkinson, K. E. 1989. *"An Introduction to Numerical Analysis"*. Wiley, New York.
- Babaud, J., Witkin, A., Baudin, M. y Duda, R. 1986. "Uniqueness of the Gaussian kernel for scale - space filtering". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:26–33.
- Baddeley, A. J. 1992a. "An error metric for binary images". En *International Workshop on Robust Computer Vision*, Bonn 9-11.
- Baddeley, A. J. 1992b. *"Robust Computer Vision: Quality of Vision Algorithms"*, cap. "An error metric for binary images", pág. 59–78. Wichmann Verlag, W. Förstner and S. Ruwiedel (eds), Karlsruhe.
- Bajcsy, M., Mintz, M. y Liebman, E. 1986. "A common framework for edge detection and region growing". *Univ. Pennsylvania GRASP Lab., Tech. Rep.*, 61.
- Ballard, D. H. 1981. "Strip trees: a hierarchical representation of curves". *Communications of the ACM*, 24:310–321.
- Ballard, D. H. y Brown, C. M. 1981. "Generalizing the Hough transform to detect arbitrary shapes". *Pattern Recognition*, 13:111–122.
- Ballard, D. H. y Brown, C. M. 1982. *"Computer Vision"*. Prentice Hall, Englewood Cliffs, N. J.
- Barranco-López, V. 1998. *"Detección de bordes en Imágenes Texturadas y Ruidosas Mediante la Divergencia de Jensen-Shannon"*. Tesis Doctoral, Departamento de Física Aplicada, Universidad de Granada, Granada, España.
- Bartels, R., Beatty, J. y Barsky, B. 1991. *"An Introduction to Splines for use in Computer Graphics and Geometric Modeling"*. Brooks/Cole.
- Basseville, M. 1981. "Edge detection using sequential methods for change in level". *IEEE Transactions on Acoustic, Speech and Signal Processing*, ASSP-29(1):32–50.
- Basseville, M., Espiau, B. y Gasnier, J. 1981. "Edge detection using sequential methods for change in level". *IEEE Transactions on Acoustic, Speech and Signal Processing*, ASSP-29(1):24–31.
- Bebis, G., Papadourakis, G. y Orphanoudakis, S. 1999. "Curvature scale-space-driven object recognition with a indexing scheme based on artificial neural networks". *Pattern Recognition*, 32:1175–1201.

- Belkasim, S. O., Shridhar, M. y Ahmadi, M. 1991. "Pattern recognition with moments invariants: a comparative study and new results". *Pattern Recognition*, 24:1117–1138.
- Bengtsson, A. y Eklundh, J. 1991. "Shape representation by multiscale contour representation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:85–93.
- Bennet, J. R. y McDonald, J. S. 1975. "On the measurement of curvature in a quantized environment". *IEEE Trans. Comput.*, 24:803–820.
- Bergholm, F. 1987. "Edge focusing". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):726–741.
- Bernsen, J. A. C. 1991. "An objective and subjective evaluation edge detection methods in images". *Philips Journal of Research*, 46:57–94.
- Bhandarkar, S. M., Zhang, Y. Q. y Potter, W. D. 1994. "An edge-detection technique using genetic algorithm-based optimization". *Pattern Recognition*, 27(9):1159–1180.
- Bigun, J. 1993. "Unsupervised feature reduction in image segmentation by local transforms". *Pattern Recognition Letters*, 14:573–583.
- Billingsley, F. C. 1979. "*Topics in Applied Physics: Picture Processing and Digital Filtering*", cap. "Noise considerations in digital image processing hardware", pág. 249–281. Springer-Verlag, second edición.
- Blum, H. 1967. "A transformation for extracting new descriptor of shape". En Whate-Dunn, W., editor, *Symposium Models for Speech and Visual Form*, pág. 103–104, Cambridge, MA. MIT Press.
- Bousquet, P. y Flouzat, G. 1986. "Segmentation by mathematical morphology processing hand on the image understanding model in remote sensing". *ESA Proceedings of the 1986 International Geoscience and Remote Sensing Symposium (IGARSS'86) on Remote Sensing: Today's Solutions for Tomorrow's Information Needs*, 1:553–557.
- Bracho, R. y Sanderson, A. C. 1985. "Segmentation of images based on intensity gradient information". En *CVPR-85 Conference on Computer Vision and Pattern Recognition*, pág. 341–347, San Francisco.
- Brink, A. D. 1995. "Minimum spatial entropy threshold selection". *IEEE Proceedings Vision, Image and Signal Processing*, 142(3):128–132.
- Bryant, D. J. y Bouldin, D. W. 1979. "Evaluation of edge operators using relatively and absolute grading". *Proceedings of the IEEE Comp. Soc. Conf. on Pattern Recognition and Image Processing*, pág. 138–145.

- Burns, J. B. y Hanson, A. R. 1986. "Extracting straight lines". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:425–455.
- Canhan, P. B. 1970. "The minimum energy of bending as a possible explanation of the biconcave shape of human red blood cell". *Journal of Theoretical Biology*, 26:61–81.
- Canny, J. F. 1983. "Finding Edges and Lines in Images". Technical Report 720, M.I.T. Artificial Intelligence Laboratory, Cambridge, MA.
- Canny, J. F. 1986. "A computational approach to edge detection". *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.
- Carron, T. y Lambert, P. 1994. "Color edge detector using jointly hue, saturation and intensity". En *Proc. IEEE International Conference on Image Processing*, pág. 977–981.
- Castan, S., Zhao, J. y Shen, J. 1990. "Optimal filter for edge detection methods and results". En *Proceedings of the First European Conference on Computer Vision (ECCV)*, pág. 13–17.
- Castleman, K. R. 1979. *"Digital Image Processing"*. Signal Processing Series. Prentice Hall.
- Cazorla, M. A., colomina, O., Escolano, F., Gallardo, D., Rizo, R. y Satorre, R. 1997. *"Técnicas de Inteligencia Artificial"*. Club Universitario.
- Chakarborty, A. y Duncan, J. S. 1999. "Game theoretic integration for image segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):12–30.
- Chan, W., Coghill, G. y Sivaswamy, J. 2001. "A simple mechanism for curvature detection". *Pattern Recognition*, 22:731–739.
- Chapron, M. 1992. "A new chromatic edge detector used for color image segmentation". En *Proc. 11th IEEE International Conference on Pattern Recognition. Conf. C: Image, Speech and Signal Analysis*, vol. III, pág. 311–314, Den Hague, Netherlands.
- Chapron, M. 1997. "A chromatic contour detector based on abrupt change techniques". En *International Conference on Image Processing*, vol. III, pág. 18–21.
- Chen, C.-S. y Tsai, W.-H. 1990. "A new fast one-pass thinning algorithm and its parallel hardware implementation". *Pattern Recognition Letters*, 11:471–477.

- Chen, W. F., Lu, X. Q., Chen, J. J. y Wu, G. X. 1995. "A new algorithm of edge - detection for color image - Generalized fuzzy operator". *Sciences in China Series A - Mathematics, Physics, Astronomy & Technological Sciences*, 38(10):1272-1228.
- Chen, Y.-S. y Yu, Y.-T. 1996. "Thinning approach for noisy digital patterns". *Pattern Recognition*, 29(11):1847-1862.
- Cheng, K. S., Lin, J. S. y Mao, C. W. 1996. "The application of competitive Hopfield neural network to medical image segmentation". *IEEE Transactions on Medical Imaging*, 15(4):560-567.
- Chi, Z. y Yan, H. 1993. "Map image segmentation based on thresholding and fuzzy rules". *Electronics Letter*, 29(21):1841-1843.
- Chidiac, H. y Ziou, D. 1999. "Clasiffication of image edges". En *Vision Interface'99*, pág. 17-24, Trois-Rivières, Canada.
- Cho, K., Meer, P. y Cabrera, J. 1997. "Performance assessment through bootstrap". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(11):1185-1198.
- Chomsky, N. 1956. "Three models for the description of language". *IRE Transactions on Information Theory*, IT-2(3):113-124.
- Chuang, G. C. H. y Kuo, C. C. J. 1996. "Wavelet descriptor of planar curves: theory and applications". *IEEE Transactions on Image Processing*, 5(1):56-70.
- Chung, P., Sai, T. y Sun, Y. 1994. "Polygonal approximation using a competitive Hopfield neural network". *Pattern Recognition*, 27:1505-1512.
- Clark, J. J. 1988. "Singularity theory and phantom edges in scale space". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):720-727.
- Clark, J. J. 1989. "Authenticating edges produced by zero-crossings algorithms". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):43-57.
- Claxton, P. R. y Kwok, E. W. 1987. "The use of colour to segment and label images". En *Alwey Vision Conference*, pág. 295-302.
- Clinque, L., Guerra, C. y Levialdi, C. 1994. "Reply: On the paper by R. M. Haralick". *CVGIP: Image Understanding*, 60(2):250-252.
- Cohen, F. S., Huang, Z. y Yang, Z. 1995. "Invariant matching and identification of curves using B-splines curve representation". *IEEE Transactions on Image Processing*, 4:1-10.

- Cong, G. y Ma, S. 1998. "Corner enhancement in curvature scale space". *Pattern Recognition*, 31(10):1491–1501.
- Cooper, J., Venkatesh, S. y Kitchen, L. 1993. "Early jump-out corner detectors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):823–828.
- Cotroulos, C. y Pitas, I. 1994. "Adaptive nonlinear filter for digital signal/image processing". En Leondes, C. T., editor, *Advances In 2D and 3D Digital Processing, Techniques and Applications*, vol. 67, pág. 263–317. Academic Press.
- Cumani, A. 1991. "Edge detection in multispectral images". *CVGIP: Graphical Model and Image Processing*, 53(1):40–51.
- Cumani, A., Grattoni, P. y Guiducci, A. 1991. "An edge-based description of color images". *CVGIP: Graphical Model and Image Processing*, 53(4):313–323.
- Daily, M. J. 1989. "Color image segmentation using Markov random fields". En *CVPR: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pág. 304–312, San Diego, Ca., USA.
- Davies, D. L. 1978. "Edge Detection in Digital Images using Small Masks and Vectors Operator and the Method of Polar Histograms". Tesis Doctoral, Department of Electrical Engineering University of Tennessee, Knoxville.
- Davies, E. R. 1990. "Machine Vision: Theory, Algorithms, Practicalities". Academic Press.
- Davies, E. R. y Plummer, A. P. N. 1981. "Thinning algorithms: a critique and new methodology". *Pattern Recognition*, 14(1-6):53–63.
- Davis, E. R. 1990. "Machine Vision: Theory, Algorithms, Practicalities". Academic Press, Madrid, España.
- Davis, L. 1975. "A survey of edge detection techniques". *Computer Graphics and Image Processing*, 4:248–270.
- Davis, L. S. 1977. "Understanding shape: angles and sides". *IEEE Transactions on Computing*, 26(3):236–242.
- Davis, L. S. 1986. "Handbook of Pattern Recognition and Image Processing", cap. "Two-Dimensional Shape Representation", pág. 233–245. Editores: Y. Young and K. S Fu. Academic Press, London.
- de Boor, C. 1978. "A Practical Guide to Splines". Springer, Berlin.
- Delcroix, C. 1988. "Fusion of edge maps in color images". En *Proc. SPIE Visual Communication Image Processing*, vol. 1001, pág. 545–554.

- Dellepiane, S., Guisto, D. D. y Vernazza, G. 1989. "Automatic parameter computation for edge detection by zero-crossing method". En *12th Conference on Signal Processing and Images (GRETI)*, pág. 617–620, France.
- Deng, W. A. y Iyengar, S. S. 1996. "A new probabilistic relaxation scheme and its application to edge - detection". *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 18(4):432–437.
- Deriche, R. 1987. "Using Canny's criteria to derive a recursive implemented optimal edge detector". *The International Journal of Computer Vision*, 1(2):167–187.
- Deriche, R. y Blaszk, T. 1993. "Recovering and characterizing image features using an efficient model based approach". En *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, USA.
- Djurić, P. M. y Fwu, J.-K. 1997. "On the detection of edges in vector images". *IEEE Transactions on Image Processing*, 6(11):1595–1601.
- Dougherty, E. R. y Astola, J. 1994. *An Introduction to Nonlinear Image Processing*. SPIE Press, Bellingham, WA., USA.
- Dougherty, S. y Bowyer, K. W. 1998. *Empirical Evaluation Techniques in Computer Vision*, cap. "Objective evaluation of edge detectors using a formally define framework", pág. 211–234. Kevin W. Bowyer and P. Jonathon Philips, IEEE Computer Society Press, Los Alamitos, CA.
- Drewniok, C. 1994. "Multi-spectral edge detection. Some experiments on data from Landsat-TM". *International Journal of Remote Sensing*, 15(18):3743–3765.
- Duda, R. O. y Hart, P. E. 1972. "Use of the Hough transformation to detect lines and curves in pictures". *Communications of ACM*, 15(1):11–15.
- Duda, R. O. y Hart, P. E. 1973. *Pattern Classification and Scene Analysis*. Wiley, New York.
- Elder, J. H. y Zucker, S. W. 1996. "Scale space localization, blur and contour based image coding". En *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pág. 27–34, San Francisco.
- Elder, J. H. y Zucker, S. W. 1998. "Local scale control for edge detection and blur estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716.
- Eom, K. B. 1998. "Shape recognition using spectral features". *Pattern Recognition Letters*, 19:189–195.
- Escalera, A. D. L. 2001. *Visión por computador*. Prentice Hall.

- Fairney, D. y Fairney, P. T. 1994. "On the accuracy of point curvature estimators in a discrete enviroment". *Image and Vision Computing*, 12(5).
- Farag, A. A. y Delp, E. J. 1995. "Edge linking by sequential search". *Pattern Recognition*, 28(5):611–633.
- Faugeras, O. 1993. "*Three Dimensional Computer Vision. A Geometric Viewpoint*". The MIT Press, Cambridge, Massachusetts; London, England.
- Feechs, R. J. y Arce, G. R. 1987. "Multidimensional morphologic edge detection". En *Conf. Visual Communication and image Processing, Proc. SPIE*, vol. 845, pág. 285–292.
- Fernández-García, N. L., Hervás-Martínez, C. y Ventura-Soto, S. 1999. "Skeletonization and thinning algorithms using neural networks: a parallelism study". En *Actas del VIII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, pág. 103–104, Bilbao.
- Fischler, M. A. y Bolles, R. C. 1986. "Perceptual Organization and curve partitioning". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):100–105.
- Fischler, M. A., Tenenbaum, J. M. y Wolf, H. C. 1981. "Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique". *Computer Vision, Graphics and Image Processing*, 5(15):201–223.
- Flusser, J. y Suk, T. 1993. "Pattern recognition by affine moment invariants". *Pattern Recognition*, 26:167–174.
- Foley, J. D., van Dam, A., Feiner, S. K. y Hughes, J. F. 1996. "*Computer Graphics. Principles and Practice. Second Edition in C*". The Systems Programming Series. Addison Wesley.
- Fotinos, A., Economou, G. y Fotopoulos, S. 1999. "Use of relative entropy in color edge detection". *Electronics Letter*, 35(18):1532–1534.
- Fram, J. R. y Deustsch, E. S. 1975. "On the quantitative evaluation of edge detection schemes and their comparison with human performance". *IEEE Transactions on Computers*, C-24(6):616–628.
- Freeman, H. 1961. "On the encoding of arbitrary geometric configuration". *IRE Transactions on Electronics Computers*, EC-10(2):260–268.
- Freeman, H. 1974. "Computer processing of line-drawing images". *Comput. Surveys*, 6.
- Freeman, H. 1978. "Shape description via the use of critical point". *Pattern Recognition*, 10:159–166.

- Freeman, H. y Saghri, J. A. 1980. "Comparative analysis of line-drawing modeling schemes". *Computer Graphics and Image Processing*, 12:203–223.
- Frei, W. y Chen, C. C. 1977. "Fast boundary detection: a generalization and a new algorithm". *IEEE Transactions on Computers*, C-26(10):988–998.
- Friel, N. y Molchanov, I. S. 1999. "A new thresholding technique based on random sets". *Pattern Recognition*, 32(9):1507–1517.
- Fries, R. W. y Modestino, J. W. 1977. "An Empirical Study of Selected Approaches to the Detection of Edges in Noisy Digitized Images". Technical Report TR 77-1, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute.
- Fu, A. M. N. y Yan, H. 1997a. "A curve bend function based method to characterize contour shapes". *Pattern Recognition*, 30(10):1661–1671.
- Fu, A. M. N. y Yan, H. 1997b. "Effective classification of planar shapes based on curve segment properties". *Pattern Recognition Letters*, 18:55–61.
- Fu, A. M. N. y Yan, H. 1997c. "Object recognition based on contour feature and recognition by a Hopfield - Amari network". *Neurocomputing*, 16:127–138.
- Fu, K. S. y Mui, J. K. 1981. "A survey on image segmentation". *Pattern Recognition*, 13:3–16.
- Fuertes, J. M., Lucena, M., de la Blanca, N. P. y Valdicia, J. F. 1997. "A new method to segmenting color images". En *VII National Symposium on Pattern Recognition and Image Analysis*, vol. I, pág. 233–238.
- Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, New York. 400-438.
- Gallego-Ayllón, C. 1998. "Implementación de algoritmos de adelgazamiento de bordes y esqueletización de regiones mediante técnicas de redes neuronales". Technical report, Escuela Politécnica Superior. Universidad de Córdoba, España. Proyecto de fin de carrera. Directores: N. L. Fernández-García y S. Ventura-Soto.
- Gambotto, J.-P. 1993. "A new approach to combining region growing and edge detection". *Pattern Recognition Letter*, 14(November):869–875.
- Gauch, J. y Hsia, C.-W. 1992. "A comparison of three color image segmentation algorithms in four color spaces". *Visual Communications and Image Processing*, 1818:1168–1181.
- Geman, D. 1988. "Random field and inverse problems imaging". *Lectures Note in Mathematics*, pág. 113–193. The Saint Flour Lectures.

- Geman, S. y Geman, D. 1984. "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-6:721-741.
- Giraudon, G. 1985. "Edge detection from local negative maximum of second derivative". En *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pág. 643-645, San Francisco, California, USA.
- Glasbey, C. A. 1993. "An analysis of histogram-based thresholding algorithms". *CVGIP: Graphical Model and Image Processing*, 55(6):532-537.
- Goetcherian, V. 1980. "From binary to gray tone image processing using fuzzy logic concepts". *Pattern Recognition*, 12:7-15.
- Gokmen, M. y Li, C. C. 1993. "Edge detection and surface reconstruction using refined regularization". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):492-499.
- Gong, W. y Bertrand, G. 1990. A note on "Thinning of 3-D images using the safe point thinning algorithm (SPTA)". *Pattern Recognition Letter*, 11:499-500.
- González, R. C. y Wints, P. 1978. *"Digital Image Processing"*. Addison - Wesley, Reading, Mass.
- González, R. C. y Woods, R. E. 1996. *"Tratamiento Digital de Imágenes"*. Addison-Wesley Iberoamericana, S. A. y Ediciones Díaz de Santos, S. A.
- González-Jiménez, J. 2000. *"Visión por Computador"*. Paraninfo, Madrid, España.
- Granlund, G. H. 1972. "Fourier preprocessing for hand print character recognition". *IEEE Trans. Comput.*, 21:195-201.
- Grassman, H. 1853. "Zur Theorie der Farbenmischung, Poggendorf". *Ann. Phy.*, 89(69):574-578. (También: *Phil. Mag.* Vol 4, No. 7, 254, 1.853).
- Grau, A. 1997. "Template masks for real time edge thinning". En *VII National Symposium on Pattern Recognition and Image Analysis*, vol. II, pág. 22-23.
- Gu, Y.-H. y Tjahjadi, T. 2000. "Coarse-to-fine planar object identification using invariant curve features and B-spline modeling". *Pattern Recognition*, 33:1411-1422.
- Guiducci, A. 1988. "Corner characterization by differential geometry techniques". *Pattern Recognition Letters*, 8(5):311-318.
- Guil, N., González-Linares, J. M. y Zapata, E. L. 1999. "Bidimensional shape detection using an invariant approach". *Pattern Recognition*, 32:1025-1038.

- Hadamard, J. 1923. *"Lectures on the Cauchy Problem in Linear Partial Differential Equations"*. Yale University Press.
- Haddon, J. F. 1987. "Generalised Threshold Selection for Edge Detection". Technical Report RAE-TR-87035;RAE-AW-24, Royal Aircraft Establishment, Farnborough, England. Sponsoring Organization: National Aeronautics and Space Administration, Washington, DC.
- Haddon, J. F. y Boyce, J. F. 1993. "Cooccurrence matrices for image analysis". *Pattern Recognition*, 5(2):71–83.
- Haralick, R. M. 1980. "Edge and region analysis for digital image data". *Computer Graphics and Image Processing*, 12:60–73.
- Haralick, R. M. 1983a. "Ridge and valley on digital images". *Computer Vision, Graphics and Image Processing*, 22:28–38.
- Haralick, R. M. 1983b. "The topographical primal sketch". *Int. J. Robot. Res.*, 2:50–72.
- Haralick, R. M. 1984. "Digital step edge from zero crossings of second directional derivatives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):56–68.
- Haralick, R. M. 1994. "Dialogue: performance characterization in computer vision". *CVGIP: Image Understanding*, 60(2):245–265.
- Haralick, R. M. y Shanmugan, K. 1973. "Computer classification of reservoir sandstones". *IEEE Transactions on Geoscience Electronics*, GE-11(4):171–177.
- Haralick, R. M. y Shapiro, L. G. 1985. "Image segmentation techniques". *Computer Vision, Graphics and Image Processing*, 29:100–132.
- Haralick, R. M. y Shapiro, L. G. 1992. *"Computer and Robot Vision"*, vol. 1,2. Addison-Wesley, Reading, Ma.
- Hardie, R. C. y Arce, G. R. 1991. "Ranking in R^P and its use in multivariate image estimation". *IEEE Transactions on Circuits and Systems for Video Technology*, 1(2):678–689.
- Hasegawa, J., Kubota, H. y Toriwaki, J. 1996. "Automated construction of image processing procedures by sample - figure representation". En *Proceedings of the 8th International Conference on Pattern Recognition*, págs. 586–588, Paris.
- He, D.-C. y Wang, L. 1992. "Detecting texture edges from images". *Pattern Recognition*, 25(6):595–600.
- Healey, G. 1992. "Segmenting images using normalized color". *IEEE Transactions on Systems, Man and Cybernetics*, 22(1):64–73.

- Hearn, D. y Baker, M. P. 1994. “*Gráficas por computadora*”. Prentice Hall, segunda edición.
- Heath, M., Sarkar, S., Sanocki, T. y Bowyer, K. 1996. “Comparison of edge detectors. A methodology and initial study”. *Computer Vision and Pattern Recognition*.
- Heath, M., Sarkar, S., Sanocki, T. y Bowyer, K. 1997. “A robust visual method for assesing the relative performance of edge detectors algorithms”. *IEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1338–1359.
- Heath, M., Sarkar, S., Sanocki, T. y Bowyer, K. 1998. “Comparison of edge detectors. A Methodology and initial study”. *Computer Vision and Image Understanding*, 69(1):38–54.
- Hedley, M. y Yan, H. 1992. “Segmentation of color images using spatial and color space information”. *Journal of Electronics Imaging*, 1:374–380.
- Henderson, T. 1983. “*Fundamental of Computer Vision*”, cap. “Syntactic and structural methods I and II”, pág. 273–292. Cambridge University Press, Cambridge.
- Hering, E. 1977. “*Zur Lehe vom Lichtsinne*”. C. Gerond’s Sohn, Vienna, Austria.
- Hewitt, F. C., Laferty, D. R. y Mahran, H. E. 1990. “Applications of mathematical morphology in imaging. 2. Advanced scene segmentation”. *American Laboratory*, 22(12).
- Hildreth, E. C. 1983. “The detection of intesisty changes by computer and biological vision systems”. *Computer Vision, Graphics and Image Processing*, 22:1–27.
- Hildreth, E. C. 1992. “Edge and local feature detection”. En “*Encyclopedia of Artificial Intelligence*”, vol. 1, pág. 422–434. Wiley Interscience Publication, John Wiley & Sons, Inc Editor: S. C. Shapiro., segunda edición.
- Holla, K. 1982. “Opponent colors as a 2-dimensional feature within a model of the first stages of the human visual system”. En *Proceedings of the 6th Int. Conf. on Pattern Recognition*, vol. 1, pág. 161–163.
- Horn, B. K. P. 1986. “*Robot Vision*”. The MIT Electronical and Computer Science Series. McGraw-Hill Book Company.
- Hough, P. V. C. 1962. “Methods and means for recognizing complex patterns”. *U. S. Patent 3,069,654*.
- Hu, M. K. 1962. “Visual pattern recognition by moment invariants”. *IRE Transactions on Information Theory*, 8:179–187.

- Huang, K. y Yan, H. 1995. "On line signature verification based on dynamic and global and local matching". *Optical Engineering*, 34(12):3480–3487.
- Huang, P. W., Dai, S. K. y Lin, P. L. 1999. "Planar shape recognition by directional flow-change method". *Pattern Recognition Letters*, 20:163–170.
- Hueckel, M. 1971. "An operator which locates edges in digital pictures". *J. Ass. Comput. Mach.*, 18:113–125.
- Hueckel, M. 1973. "A local visual operator which recognizes edges and lines". *J. Ass. Comput. Mach.*, 20:634–647.
- Hueckel, M. 1974. Erratum for "A local visual operator which recognizes edges and lines". *J. Ass. Comput. Mach.*, 21:350.
- Huertas, A. y Medioni, G. 1986. "Detection of intensity changes with subpixel accuracy using laplacian gaussian mask". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):651–664.
- Huntsberger, T. L. y Descalzi, M. L. 1985. "Color edge detection". *Pattern Recognition Letters*, 3:205–209.
- Huntsberger, T. L., Jacobs, C. L. y Cannon, R. L. 1985. "Iterative fuzzy image segmentation". *Pattern Recognition*, 18:131–138.
- IC-PCI 1995. "*MVC-ICTM IC-PCI Hardware Reference Manual*". Imaging Technology.
- Ikebe, Y. y Miyamoto, S. 1982. "Shape design, representation and restoration with splines". En Fu, K. y Kunii, T., editores, *Picture Engineering*, págs. 75–95, Berlin. Springer.
- Iyengar, S. S. y Deng, W. 1995. "An efficient edge detection algorithm using relaxation labeling technique". *Pattern Recognition*, 28(4):519–536.
- Jähne, B. 1993. "*Digital Image Processing. Concepts, Algorithms and Scientific Applications*". Springer-Verlag, segunda edición.
- Jain, A. K. 1981. "Advances in mathematical model for image processing". En *Proceedings of the IEEE*, vol. 69, págs. 502–528.
- Jain, A. K. 1989. "*Fundamentals of Digital image Processing*". Information and System Sciences Series. Prentice Hall.
- Jayaramamurthy, S. N. y Huntsberger, T. L. 1985. "Edge and region analysis using fuzzy sets". En *Proceedings of the IEEE Workshop on Language and Automation*, págs. 71–75.
- Jeong, H. y Kim, C. I. 1992. "Adaptative determination of filter scale for edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, págs. 579–585.

- Jiang, X. Y., Hoover, A., Jean-Baptiste, G., Goldgof, D., Bowyer, K. y Bunke, H. 1995. "A methodology for evaluating edge detection algorithms". En *Proceedings of Asian Conference on Computer Vision*, pág. 415–419.
- Kanungo, T., Jaisimha, Y., Palmer, J. y Haralick 1995. "A methodology for quantitative performance evaluation of detection algorithms". *IEEE Transactions on Image Processing*, 4(12).
- Kapur, J. N., Sahoo, P. K. y Wong, A. K. C. 1985. "A new method for gray-level picture thresholding using the entropy of the histogram". *Computer Vision, Graphics and Image Processing*, 29:273–285.
- Kerfoot, I. B. y Bresler, Y. 1999. "Theoretical-analysis of multispectral image segmentation criteria". *IEEE Transactions on Image Processing*, 8(6):798–820.
- Kholmann, K. 1996. "Corner detection in natural images based on the 2-D Hilbert transform". *Signal Processing*, 48:225–234.
- Khotanzad, A. y Chen, J.-Y. 1989. "Unsupervised segmentation of textured images by edge detection in multidimensional features". *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 11(4):414–421.
- Kim, H. J., Kim, E. Y., Kim, J. W. y Park, S. H. 1998. "Mrf model - Based image segmentation using hierchical distributed genetic algorithm". *Electronics Letters*, 34(25):2394–2395.
- Kincaid D. y Cheney, W. 1991. "*Numerical Analysis*". Brooks/Cole.
- Kirsch, R. 1971. "Computer determination of the constituent structure of biological images". *Compt. Biomed. Res.*, 4(3):315–328.
- Kiryati, N. y Maydan, D. 1989. "Calculating geometric properties form Fourier representacion". *Pattern Recognition*, 22(5):469–475.
- Kisworo, M., Venkatesh, S. y West, G. 1994. "Modeling edges at subpixel accuracy using the local energy approach". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):405–410.
- Kitchen, L. y Malin, J. A. 1989. "The effect of spatial discretization on the magnitude and direction response of simple differential edge operators on a step edge". *Computer Vision, Graphics and Image Processing*, 47:243–258.
- Kitchen, L. y Rosenfeld, A. 1981. "Edge evaluation using local edge coherence". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(9):597–605.
- Kittler, J. y Illingworth, J. 1985. "On threshold selection using clustering criteria". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15(5):652–655.

- Korn, A. F. 1988. "Toward a symbolic representation of intensity changes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):610–625.
- Koschan, A. 1995. "A comparative study on color edge detection". En *Proceedings 2nd Asian Conference on Computer Vision ACCV'95*, vol. III, pág. 574–578.
- Koundinya, K. N. V. L. N. y Chanda, N. 1994. "Detecting lines in gray level images using search techniques". *Signal Processing*, 37:287–299.
- Krishnamoorthi, R. y Bhattacharyya, P. 1997. "Color edge detection using orthogonal polynomials". En *Lectures Notes in Computer Sciences*, vol. 1351, pág. 590–597.
- Krishnapuram, R. y Chen, L.-F. 1990. "Iterative neural networks for skeletonization and thinning". *Intelligent Robots and Computer Vision*, IX:271–281.
- Krishnapuram, R. y Chen, L.-F. 1993. "Implementation of parallel thinning algorithms using recurrent neural networks". *IEEE Transactions on Neural Networks*, 4(1):142–147.
- Krishnapuram, R. y Gupta, S. 1992. "Morphological methods for detection and classification of edge in range images". *Journal of Mathematical Imaging and Vision*, 2:351–375.
- Lacroix, V. 1990. "Edge detection: what about rotation invariance". *Pattern Recognition Letters*, 11:797–802.
- Lam, L., Lee, S. W. y Suen, C. Y. 1992. "Thinning methodologies - A comprehensive survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):868–885.
- Lanser, S. 1993. "Kantenorientierte Farbsegmentation im Cie-Lab Raum". En Pöpl, S. J. y Handel, H., editores, *Proc. 15 DAGM-Symposium Mustererkennung*, pág. 639–646, Lübeck, Germany. (In German).
- Lanser, S. y Eckstein, W. 1992. "A modification of Deriche's approach to edge detection". En *Proc. 11th Int. Conf. On Pattern Recognition*, vol. III de *Conf. C: image, Speech and Signal Analysis*, pág. 633–637, Den Hague, neherlands.
- Lee, D., Wasilkowski, G. W. y Mehrotra, R. 1993. "A new zero crossing based discontinuity". *IEEE Transactionson on Pattern Analysis and Machine Intelligence*, 2:265–268.
- Lee, H. C. y Cok, D. R. 1991. "Detecting boundaries in vector field". *IEEE Transactions on Signal Processing*, 39:1181–1194.
- Lee, J. S. 1981. "Refined filtering of image Noise using local statistics". *Computer Vision Graohics Image Processing*, 15:380–389.

- Lee, J. S. y Hoppel, K. 1989. "Noise modelling and estimation of remotely-sensed images". *Proceedings of 1989 International Geoscience and Remote Sensing*, 2:1005–1008.
- Lee, J. S. J., Haralick, R. M. y Shapiro, L. G. 1987. "Morphologic edge detection". *IEEE Journal of Robotics and Automation*, RA-3(2):142–156.
- Lee, S. U. y Chung, S. Y. 1990. "A comparative performance study of several global thresholding techniques for segmentation". *Computer Vision, Graphics and Image Processing*, 52:171–190.
- Lemoigne, J. y Tilton, J. C. 1995. "Refining image segmentation by integration of edge and region Data". *IEEE Transactions on Geoscience and Remote Sensing*, 33(3):605–615.
- Leung, C. K. y Lam, F. K. 1996. "Performance analysis for a class of iterative image thresholding algorithms". *Pattern Recognition*, 29(9):1523–1530.
- Levine, M. D. 1985. *"Vision in Man and Machine"*. Mc Graw-Hill.
- Li, C. H. y Lee, C. K. 1993. "Minimum cross entropy thresholding". *Pattern Recognition*, 26(4):617–625.
- Li, M. y Wu, P. S. 1997. "Pyramid edge detection for color images". *Optical Engineering*, 36(5):1431–1437.
- Lin, C. C. y Chellapa, R. 1987. "Classification of partial 2D shapes using Fourier descriptors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:686–690.
- Lin, Y., Dou, J. y Wang, H. 1992. "Contour shape description based on an arc height function". *Pattern Recognition*, 25:17–23.
- Lindeberg, T. 1993. "On scale selection for differential operators". En *Proceedings of 8th Scandinavian Conference on Image Analysis*, pág. 857–866, Tromsø, Norway.
- Lindeberg, T. 1994. "Junction detection with automatic selection of detection scales and localization scales". En *Proceedings of IEEE, International Conference on Image Processing and Pattern Recognition*, pág. 924–928, Austin, USA.
- Lindeberg, T. 1996. "Edge detection and ridge detection with automatic scale selection". En *Proceedings of IEEE, International Conference on Image Processing and Pattern Recognition*, vol. 11(4), pág. 337–356.
- Liscano, R. y Wong, A. K. C. 199x. "A Study into Entropy-based Thresholding for Image Edge Detection". Technical report, Autonomous Systems Laboratory, National Research Council of Canada. Department of System Design, University of Waterloo. e-mail:liscano@iit.ncr.ca.

- Liu, H. y Srinath, M. 1990. "Partial shape classification using contour matching in distance transformation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:1072–1079.
- Loncaric, S. 1998. "A survey of shape analysis techniques". *Pattern Recognition*, 31(8):983–1001.
- Lu, Y. y Jain, R. C. 1989. "Behavior of edges in scale space". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(4):337–356.
- Lu, Y. y Jain, R. C. 1992. "Reasoning about edges in scale space". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):450–468.
- Lunscher, W. H. H. J. y Beddoes, M. P. 1986. "Optimal edge detector desing I: parameter selection and noise effects". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):164–177.
- Luong, Q. T. 1991. "La couleur en vision par ordinateur: une revue". *Traitement du Signal*, 8:548–555.
- Lyvers, E. P. y Mitchell, O. R. 1988. "Precision edge constrast and orientation estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):927–937.
- Macaire, L., Ultré, V. y Postaire, J.-G. 1996. "Determination of compatibility coefficients for color edge detection by relaxation". En *International Conference on Image Processing*, vol. 19P9, pág. 1045–1048.
- Machuca, R. y Philips, K. 1983. "Applications of vector fields to image processing". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):316–329.
- Maragos, P. 1996. "Diferential morphology and image processing". *IEEE Transactions on Image Processing*, 5(6):922–937.
- Maragos, P. y Schafer, R. W. 1990. "Morphological systems for multidimensional signal processing". *Proceedings of the IEEE*, 78(4):690–710.
- Maravall, D. 1993. *Reconocimiento de Formas y Visión Artificial*. Ediciones Ra-Ma.
- Marr, D. 1982. *Vision - A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H, Freeman and Co., San Francisco.
- Marr, D. y Hildreth, E. C. 1980. "Theory of edge detection". *Proceedings of the Royal Society of London B207*, pág. 1223–1233.
- Martelli, A. 1972. "Edge detection using heuristic search methods". *Computer Graphics and Image Processing*, 1:169–182.

- Matsuyama, T. 1989. "Expert system for image processing: knowledge - based composition of image analysis processes". *Computer Vision, Graphics and Image Processing*, 48:22-49.
- Meer, P., Jolion, J. y Rosenfeld, A. 1990. "A fast parallel algorithm for blind estimation of noise variance". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):216-223.
- Mehrotra, R., Nichani, S. y Ranganathan, N. 1990. "Corner detection". *Pattern Recognition*, 23:1223-1233.
- Meyer, F., A.Oliveras, Salembier, P. y Vachier, C. 1997. "Morphological tools for segmentation - connected filters and watersheds". *Annales Des Telecommunications - Annals of Telecommunications*, 52(7-8):367-379.
- Michelli, E. D., Caprile, B., Ottonello, P. y Torre, V. 1989. "Localization and noise in edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1106-1117.
- Modestino, J. W. y Fries, R. W. 1977. "Edge detection in noisy images using recursive digital filtering". *Computer Graphics and Image Processing*, 6:409-433.
- Modestino, J. W. y Zhang, J. 1994. "A Markov random field model - Based approach to image interpretation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14:969-976.
- Moghaddamzadeh, A. y Bourbakis, N. 1997. "A fuzzy region growing approach for segmentation of color images". *Pattern Recognition*, 30(6):867-881.
- Mokhtarian, F. 1995. "Silhouette-based isolated object recognition through curvature scale space". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):539-544.
- Mokhtarian, F. y Mackworth, A. K. 1992. "A theory of multiscale-based shape representation for planar curves". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789-805.
- Monga, O. D., Deriche, R. y Rocchisani, J. M. 1991. "3-D edge detection using recursive filtering: application to scanner images". *CVGIP: Image Understanding*, 53(1):76-87.
- Montanari, U. 1971. "On the optimal detection of curves in noisy pictures". *Communications of the ACM*, 14(5):335-345.
- Mukherjee, J. y Chatterji, B. N. 1989. "Thinning of 3-D images usign the safe point thinning algorithm (SPTA)". *Pattern Recognition Letter*, 10:167-173.

- Nakagawa, Y. y Rosenfeld, A. 1978. "A note on the use of local max and min operations in digital picture processing". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-8:632–635.
- Nakamura, Y., Sumida, N. y Ohta, Y. 1998. "Object arrangement estimation using color edge profile". En *International Conference on Pattern Recognition*, pág. 754–758.
- Nalwa, V. S. 1987. "Edge detector resolution improvement by image interpolation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):446–451.
- Nalwa, V. S. 1993. *"A Guided Tour of Computer Vision"*. Addison-Wesley.
- Nalwa, V. S. y Binford, T. 1986. "On detecting edges". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:699–714.
- Nayar, S. K., Baker, S. y Murase, H. 1996. "Parametric feature detection". En *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pág. 471–477.
- Nevatia, R. 1977. "A color edge detector and its use in scene segmentation". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-7(11):820–826.
- Nevatia, R. 1982. *"Machine Perception"*. Prentice Hall, Englewood Cliffs, New Jersey.
- Niblack, W. 1986. *"An Introduction to Digital Image Processing"*. Prentice Hall.
- Noble, J. A. 1992. "Finding half boundaries and junctions in images". *Image and Vision Computing*, 10(4):219–232.
- Novak, C. L. y Shafer, S. A. 1987. "Color edge detection". *Proceedings DARPA Image Understanding Workshop, Los Angeles, CA, USA*, I:35–37.
- Ogawa, H. 1989. "Corner detection on digital curves based on local symmetry of shapes". *Pattern Recognition*, 22(4):351–357.
- O’Gorman, L. 1990. "K × K thinning". *Computer Vision, Graphics and Image Processing*, 51:195–215.
- Ohlander, R. 1975. *"Analysis of Natural Scenes"*. Tesis Doctoral, Department of Computer Science. Carnegie-Mellon University.
- Ohta, Y. 1985. *"Knowledge-based interpretation of outdoor natural color scenes"*. Pitman Publishing Limited.
- Ohta, Y., Kanade, T. y Sakai, T. 1980. "Color information for region segmentation". *Computer Graphics and Image Processing*, 13:222–241.

- Oliver, C. J. y Lombardo, P. 1996. "Simultaneous mean and texture edge detection in SAR clutter". *IEEE Proceeding - Radar, Sonar and Navigation*, 143(6):391–399.
- Olsen, S. I. 1993. "Estimation of noise in images: an evaluation". *Graphical models and Image Processing*, 55(4):319–323.
- Oppenheim, A. V., Schafer, R. W. y Stockham, T. G. 1968. "Nonlinear filtering of multiplied and convolved signals". *Proceedings of the IEEE*, 56:1264–129.
- Oppenheim, A. V., Willsky, A. S. y Young, I. T. 1983. *"Signal and Systems"*. Prentice Hall, Englewood Cliffs, NJ.
- Owens, R. y Venkatesh, S. 1989. "Edge detection is a projection". *Pattern Recognition Letters*, 9:233–244.
- Pal, N. R. y Pal, S. K. 1993. "A review on image segmentation techniques". *Pattern Recognition*, 26(9):1277–1294.
- Pal, S. K. 1989. "Fuzzy skeletonization". *Pattern Recognition Letter*, 10:17–23.
- Pal, S. K. y King, R. A. 1983. "On edge detection of X-ray images using fuzzy set". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:69–77.
- Palmer, P. L., Dabis, H. y Kittler, J. 1996. "A performance measure for boundary detection algorithms". *Computer Vision and Image Understanding*, 63(3):476–494.
- Panda, D. P. y Dubitzki, T. 1979. "Statistical analysis of some edge operators". *Computer Graphics and Image Processing*, 11(3):313–348.
- Papoulis, A. 1991. *"Probability, Random Variables and Stochastic"*. McGraw Hill, New York, 3rd edition edición.
- Park, R.-H. y Choi, W. Y. 1989. "A new interpretation of the compass gradient edge operators". *Computer Vision, Graphics and Image Processing*, 47:259–265.
- Parker, J. R. 1997. *"Algorithm for Image Processing and Computer Vision"*. John Wiley & Sons.
- Pavlidis, T. 1972. "Segmentation on pictures and maps through functional approximation". *Computer Vision, Graphics and Image Processing*, 1:360–372.
- Pavlidis, T. 1975. "Optimal piecewise polynomial L_2 approximations of functions of one and two variables". *IEEE Trans. Comput.*, C-24:98–102.
- Pavlidis, T. 1977. "Polygonal approximations by Newton's method". *IEEE Trans. Comput.*, C-26:800–807.

- Pavlidis, T. 1978. "A review of algorithms for shape analysis". *Computer Graphics and Image Processing*, 7:243–258.
- Pavlidis, T. 1980. "Algorithms for shape analysis of contours and waveforms". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:301–312.
- Pavlidis, T. y Chen, P. C. 1979. "Segmentation by texture using a co-occurrence matrix and a split and merge Algorithm". En *Proceedings Conference on Computer Graphics, Pattern Recognition and Data Structure*.
- Pavlidis, T. y Liow, Y.-T. 1990. "Integrating region growing and edge detection". *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 12(3):225–233.
- Peak, J. E. y Tag, P. M. 1994. "Segmentation of satellite imagery using hierarchical thresholding and neural networks". *Journal of Applied Meteorology*, 33(5):605–616.
- Peli, T. 1979. "A Study of Algorithm for Edge Detection in Images. M.Sc.Thesis". Tesis Doctoral, Technion - Israel Institute of Technology, Department of Electrical Engineering.
- Peli, T. y Malah, D. 1982. "A study of edge detection algorithms". *Computer Graphics and Image Processing*, 20:1–21.
- Pérez, F. y Koch, C. 1994. "Toward color image segmentation in analog VLSI: Algorithm and hardware". *International Journal of Computer Vision*, 12(1):17–42.
- Perkins, W. A. 1980. "Area segmentation of images using edge points". *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2:8–15.
- Perona, P. y Malik, J. 1990. "Scale - space and edge detection using anisotropic diffusion". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.
- Persoon, E. y Fu, K. 1977. "Shape discrimination using Fourier descriptors". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-7:170–170.
- Petrou, M. y Kittler, J. 1991. "Optimal edge detector for ramp edge". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):483–491.
- Pietikäinen, M. y Harwood, D. 1986. "Edge information in color images Based on histograms of differences". En *International Conference on Pattern Recognition*, pág. 594–596.

- Pietikäinen, M. y Rosenfeld, A. 1981. "Image segmentation by texture using pyramid node linking". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-18:822–825.
- Pinho, A. J. y Almeida, L. B. 1997. "Neural network classifier for edge detection". En *VII National Symposium on Pattern Recognition and Image Analysis*, vol. I, pág. 371–376.
- Pitas, I., y Venetsanopoulos, A. N. 1990. *"Nonlinear Digital Filters; Principles and Applications"*. Kluwer Academic Publishers, Boston, MA.
- Pitas, I. 1993. *"Digital Image Processing Algorithms"*. International Series in Acoustics, Speech and Signal Processing. Prentice Hall.
- Pitas, I. y Venetsanopoulos, A. N. 1986. "Edge detector based on nonlinear filter". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):538–550.
- Plataniotis, K. N., Androustos, D. y Venetsanopoulos, A. N. 1997. "Color image filters: the directional approach". *Optical Engineering*, 36(9):2375–238.
- Plataniotis, K. N. y Venetsanopoulos, A. N. 2000. *"Color Image Processing and Applications"*. Digital Signal Processing Series. Springer.
- Poggio, T., Voorhees, H. y Yuille, A. 1988. "A regularized solution to edge detection". *J. Complexity*, 4:106–123.
- Pogorelov, A. 1987. *"Geometry"*. Mir Publishers, Moscow.
- Ponce, J. y Brady, M. 1985. "Toward a surface primal sketch". En Kanade, T., editor, *Three Dimensional Vision*, New York. Academic Press.
- Pratt, W. K. 1978. *"Digital Image Processing"*. John Wiley & Sons.
- Pratt, W. K. 1991. *"Digital Image Processing"*. John Wiley & Sons, New York, second edición.
- Prewitt, J. M. S. 1970. "Object enhancement and extraction". En Lipkin, B. y Rosenfeld, A., editores, *Picture Processing and Psychophysics*, pág. 75–149, New York. Academic Press.
- Prokop, R. J. y Reeves, A. P. 1993. "A survey of moment-based techniques for unoccluded object representation and recognition". *CVGIP: Graphical Model and Image Processing*, 54:438–460.
- Puentes-Rojas, D. 1997. "Implementación en Khoros 2.1 de Algoritmos para la Segmentación Digital de Imágenes en Color mediante Técnicas de Detección de Bordes". Technical report, Escuela Politécnica Superior. Universidad de Córdoba, España. Proyecto de fin de carrera. Director: N. L. Fernández-García.

- Ramer, U. 1972. "An iterative procedure for the polygonal approximation of plane curves". *Computer Graphics and Image Processing*, 1:244–256.
- Ramesh, V. y Haralick, R. M. 1992a. "Performance characterization of edge detectors". *Applications of Artificial Intelligence X: Machine Vision and Robotics*, SPIE 1708:252–266.
- Ramesh, V. y Haralick, R. M. 1992b. "Random perturbation models & performance characterization in computer vision". En *Proceedings of the Conference on Computer Vision*, pág. 521–527.
- Rangarajan, K., Shah, M. y Brackley, D. V. 1989. "Optimal corner detector". *Computer Vision, Graphics and Image Processing*, 48:230–245.
- Rattarangsi, A. y Chin, R. T. 1992. "Scale based description of corner of planar curves". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):430–449.
- Richards, C. W. y Hemami, H. 1974. "Identification of three-dimensional objects using Fourier descriptors of boundary curve". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-4:371–378.
- Rishikesh, N. y Venkatesh, Y. V. 2001. "Shape recognition using a invariant pulse code and a hierarchical, competitive neural network". *Pattern Recognition*, 34:841–853.
- Roberts, L. G. 1965. "Machine perception of three dimensional solids". En Tippet, J. T., editor, *Optical and Electro-Optical Information Processing*, pág. 159–197, Cambridge, Ma. MIT Press.
- Robinson, G. 1976. "Color edge detection". En *Proc. SPIE Symp. Advances in Image Transmission Techniques*, vol. 87, San Diego, CA.
- Robinson, G. S. 1977a. "Color edge detection". *Optical Engineering*, 16(5):479–484.
- Robinson, G. S. 1977b. "Edge detection by compass gradient masks". *Computer Graphics and Image Processing*, 6:492–501.
- Rohr, K. 1992. "Corners by fitting parametric models". *International Journal of Computer Vision*, 9(3):213–230.
- Román-Roldán, R., Gómez-Lopera, J. F., Atae-Allah, C., Martínez-Aroza, J. y Luque-Escamilla, P. L. 2001. "A measure of quality for evaluation methods of segmentation and edge detection". *Pattern Recognition*, 34:969–980.
- Rosenberg, B. 1972. "The analysis of convex blobs". *Computer Graphics and Image Processing*, 1:183–192.

- Rosenberg, B. 1974. "Computer dominant point on simple shapes". *International Journal of Man-Machine Studies*, 6:1–12.
- Rosenfeld, A. y Johnston, E. 1973. "Angle detection on digital curves". *IEEE Transactions on Computing*, C-22:875–878.
- Rosenfeld, A. y Kak, A. C. 1982a. "*Digital Picture Processing*", vol. 2 de *Series of Computer Science and Applied Mathematics*. Academic Press, segunda edición.
- Rosenfeld, A. y Kak, A. C. 1982b. "*Digital Picture Processing*", vol. 1 de *Series of Computer Science and Applied Mathematics*. Academic Press, segunda edición.
- Rosenfeld, A. y Thurston, M. 1971. "Edge and curve detection for visual scene analysis". *IEEE Transactions on Computer*, 20(5):562–569.
- Rosenfeld, A. y Torre, P. D. L. 1983. "Histogram concavity analysis as an aid in threshold selection." *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(3):231–235.
- Rosenfeld, A. y Weszka, J. S. 1975. "Angle detection on digital curves". *IEEE Transactions on Computing*, C-24:940–941.
- Rubner, Y., Tomasi, C. y Guibas, L. J. 1998. "A metric for distributions with applications to image databases". En *ICCV:Proceedings of the IEEE Conference on Computer Vision*, pág. 59–66.
- Ruzon, M. A. y Tomasi, C. 1999. "Color edge detection with compass operator". En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pág. 160–166, Ft. Collins, CO.
- Saber, E., Tekalp, A. M. y Bozdagi, G. 1997. "Fusion of color and edge information for improved segmentation and edge linking". *Image and Vision Computing*, 15:769–780.
- Saghri, J. A. y Freeman, H. 1981. "Analysis of the precision of generalized chain codes for the representation of planar curves". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5):533–544.
- Sahoo, P. K., Soltani, S., Wong, A. K. C. y Chen, Y. C. 1985. "A survey of thresholding techniques". *Computer Vision, Graphics and Image Processing*, 41:233–260.
- Salinas, R. A., Richardson, C., Abidi, M. A. y Gonzalez, R. C. 1996. "Data fusion: color edge detection and surface reconstruction through regularization". *IEEE Transactions on Industrial Electronics*, 43(3):355–363.

- Salotti, M., Bellete, F. y Garbay, C. 1996. "Evaluation of edge detectors: critics and proposal". En Christensen, H. I., Förstner, W. y Madsen, C. B., editores, *Workshop on Performance Characteristics of Vision Algorithms (Chapter 6)*, Robin College, Cambridge. (<http://www.vision.auc.dk/~hic/perfproc.html>).
- Sarkar, S. y Boyer, K. L. 1991. "On optimal infinite impulse response edge detection filters". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1154–1171.
- Sarkar, S. y Boyer, K. L. 1994. Comments "On optimal infinite impulse response edge detection filters". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):106–108.
- Schalkoff, R. J. 1989. *Digital Image Processing and Computer Vision*. John Wiley and Sons.
- Scharcanski, J. y Venetsanopoulos, A. N. 1997. "Edge detection of color images using directional operators". *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):397–401.
- Schetzen, M. 1982. *The Volterra and Wiener Theories of Nonlinear Filters*. J. Wiley & Sons, New York, USA.
- Serra, J. 1982. *Image Analysis and Mathematical Morphology*, vol. 1. Academic Press, London.
- Shah, J. 1990. "Parameter estimation, multiscale representation and algorithms for energy-minimizing segmentation". En *International Conference On Pattern Recognition*, pág. 815–819.
- Shanmugam, K. S., Dickey, F. M. y Green, J. A. 1979. "An optimal frequency domain filter for edge detection in digital pictures". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(1):37–49.
- Shen, J. y Castan, S. 1986. "An optimal linear operator for edge detection". En *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pág. 109–114, Miami.
- Shin, M. C., Goldgof, D. y Bowyer, K. W. 1998a. "An objective comparison methodology of edge detection algorithms using a structure from motion task". *Proceedings of the IEEE Comp. Soc. Conf. on Pattern Recognition and Image Processing*, pág. 190–195.
- Shin, M. C., Goldgof, D. y Bowyer, K. W. 1998b. *Empirical Evaluation Techniques in Computer Vision*, cap. "An objective comparison methodology of edge detection algorithms using a structure from motion task", pág. 235–254. Kevin W. Bowyer and P. Jonathon Philips, IEEE Computer Society Press, Los Alamitos, CA.

- Shiozaki, A. 1986. "Edge extraction using entropy operator". *Computer Vision, Graphics and Image Processing*, 36:116–126.
- Skarbek, W. y Koschan, A. 1994. "Color image Segmentation - A survey -". Teschnischer Bericht 32, Technische Universität Berlin. Fachbereich 13 Informatik, Franklinstrasse 28/29, 10587 Berlin, Germany. (<http://cv.cs.tu-berlin.de/koschan/coseg.html>).
- Smeulders, A. W. M., Vossepoel, A. M., Vrolijk, J., Ploem, J. S. y Cornelisse, C. J. 1980. "Some shape parameters for cell recognition". En *Proceedings of Pattern Recognition in Practice*, pág. 131–142, Amsterdam. North Holland.
- Smith, S. M. 1995. "Edge Thinning Used in the SUSAN Edge Detector". Technical Report TR95SMS5, Oxford Centre for Functional Magnetic Resonance Imaging of the Brain (FMRIB), Department of Clinical Neurologa, Oxford University, Oxford, United Kingdom.
- Sobel, I. 1978. "Neighbourhood coding of binary images for fast contour following and general binary array processing". *Computer Graphics and Image Processing*, 8:127–135.
- Sobel, I. E. 1970. "Camera Models and Machine Perception". Stanford University. 277-284.
- Song, X. 1993. "Robust edge detector based on morphological filters". *Pattern Recognition Letters*, 1993(14):889–894.
- Sonka, M., Hlavac, V. y Boyle, R. 1993. "Image Processing, Analysis and Machine Vision". Chapman & Hall, primera edición.
- Sossa, J. H. 1989. "An improved parallel algorithm for thinning digital patterns". *Pattern Recognition Letters*, 10(2):77–80.
- Spacek, L. A. 1986. "Edge detection and motion detection". *Image and Vision Computing*, 4:43–56.
- Spann, M. y Horne, C. 1989. "Image segmentation using a dinamyc thresholding pyramid". *Pattern Recognition*, 22(6):719–732.
- Spreeuwers, L. J. y van der Heijden, F. 1992. "Evaluation of edge detectors using average risk". En *Proceedings of IEEE, International Conference on Pattern Recognition*, pág. 771–774.
- Srinivasan, V., Bhatia, P. y Ong, S. H. 1994. "Edge-detection using a neural-network". *Pattern Recognition*, 27:1653–1662.
- Stokman, H. y Gevers, T. 1999. "Detection and classification of hyper-spectral edges". En *BMVC British Machine Vision Conference*, pág. 643–651.

- Stokman, H. y Gevers, T. 2000a. "Classifying color transition into shadow - geometry, illumination, highlight or material edges". *Submitted to IEEE ICIP*. (<http://gene.wins.uva.nl/~gevers/pub.html>).
- Stokman, H. y Gevers, T. 2000b. "Parameter-free thresholding and classification of color edges". *Submitted to Image and Vision Computing*. (<http://gene.wins.uva.nl/~gevers/pub.html>).
- Strickland, R. N. y Cheng, D. K. 1993. "Adaptable edge quality metric". *Optical Engineering*, 32(3):944-951.
- Surka, S. y Valavanis, K. P. 1991. "A cellular automata model for edge detection". *Journal of Intelligent and Robotic Systems*, 4:379-391.
- Tabatai, A. K. y Mitchell, O. R. 1984. "Edge location to subpixel values in digital imagery". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):188-201.
- Tabbone, S. 1994a. "Detecting junctions using properties of gaussian detector". En 12th *International Conference on Pattern Recognition*, pág. 52-56, Israel.
- Tabbone, S. 1994b. "*Edge Detection, Subpixel and Junctions Using Multiple Scales*". Tesis Doctoral, Institute National Polytechnique de Lorraine, France. En francés.
- Tabbone, S. y Ziou, D. 1991. "Elimination of false edge by separation and propagation of threshold". En 13th *Conference on Signal Processing and Images (GRETI)*, pág. 1025-1028, France.
- Tabbone, S. y Ziou, D. 1992. "Subpixel positioning of edges for first and second order operators". En 11th *International Conference on Pattern Recognition*, pág. 655-658, Netherlands.
- Tabbone, S. y Ziou, D. 1993. "Efficient edge detection using two scales". En *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*.
- Tagare, H. D. y de Figueiredo, R. 1990. "On the localization performance measure and optimal edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1186-1190.
- Tagare, H. D. y de Figueiredo, R. 1994. Reply to "Comments on optimal infinite impulse response edge detection filters". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):108-110.
- Tai, W. H. 1985. "Moment - preserving thresholding: a new approach". *Computer Vision, Graphics and Image Processing*, 29:373-393.

- Tamura, H., Mori, S. y Yamawaki, T. 1978. "A Comparison of line thinning algorithms from digital geometry viewpoint". En *Proceedings of the 4th International Conference of Pattern Recognition*, pág. 715–719.
- Tao, H. y Huang, T. S. 1997. "Color image edge detection using cluster analysis". En *International Conference on Image Processing*, vol. I, pág. 834 – 836.
- Teague, M. R. 1980. "Image analysis via the general theory of moments". *J. Opt. Soc. Amer.*, 70:920–930.
- Teh, C. H. y Chin, R. T. 1989. "On the detection of dominant points on digital curves". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):859–872.
- Teuber, J. 1992. *Digital Image Processing*. International Series in Acoustics, Speech and Signal Processing. Prentice Hall.
- Torre, V. y Poggio, T. A. 1986. "On edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):147–163.
- Trahanias, P. E., Karakos, D. y Venetsanopoulos, A. N. 1996. "Directional processing of color images: theory and experimental results". *IEEE Transactions on Image Processing*, 5(6):868–880.
- Trahanias, P. E. y Venetsanopoulos, A. N. 1993. "Color edge detection using vector order statistics". *IEEE Transactions on Image Processing*, 2(2):259–264.
- Trahanias, P. E. y Venetsanopoulos, A. N. 1996. "Vector order operator as color edge detectors". *IEEE Transactions on Systems and Cybernetics - Part B: Cybernetics*, 26(1):135–143.
- Tsai, D.-M. y Chen, Y.-H. 1992. "A fast histogram clustering approach for multilevel thresholding". *Pattern Recognition Letters*, 13:245–252.
- Tsang, P. W. M. y Tsang, W. H. 1996. "Edge detection on object color". En *International Conference on Image Processing*, vol. 19P9, pág. 1049–10521.
- Tsang, W. H. y Tsang, P. W. M. 1997. "Suppression of false edge detection due to specular reflection in color images". *Pattern Recognition Letters*, 18:165–171.
- Ueda, N. y Suzuki, S. 1993. "Learning visual models from shape contours using multiscale convex/concave structure matching". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):337–352.
- Uhr, L. 1966. *Pattern Recognition: Theory, Experiments, Computer Simulations and Dynamical Models of Forms, Perception and Discovery*. Wiley, New York. (Editor).

- Ullman, S. 1981. "Analysis of visual motion by biological and computer system". *IEEE Computer*, 14(8):57–69.
- van Otterloo, P. J. 1991. "A contour-oriented approach to shape analysis". Prentice Hall, Hemel Hempstead.
- Varona, J. y Villanueva, J. J. 1997. "Neural network for early vision". En *VII National Symposium on Pattern Recognition and Image Analysis*, vol. I, pág. 377–382.
- Venkatesh, S. y Kitchen, L. J. 1992. "Edge evaluating using necessary components". *CVGIP: Graphical Models and Image Processing*, 54(1):23–30.
- Venkatesh, S. y Owens, R. 1990. "A energy feature detection scheme". En *Proceedings of IEEE, International Conference on Pattern Recognition*, pág. 553–557, Singapore.
- Vorhees, H. y Poggio, T. 1987. "Detecting textons and texture boundaries in natural images". En *Proceedings of the International Conference on Computer Vision*, pág. 250–258, London.
- Wallace, T. y Wintz, P. 1980. "An efficient three-dimensional aircraft recognition algorithm using normalized Fourier descriptors". *Computer Graphics and Image Processing*, 13:99–126.
- Wang, P. S. P. y Zhang, Y. Y. 1989. "A fast and flexible thinning algorithm". *IEEE Transactions on Computers*, 39(5):741–745.
- Wang, S.-J. 1994. "Model-based edgel aggregation". *ARPA Image Understanding Workshop Proceedings*.
- Weiss, I. 1993. "Geometric invariants and object recognition". *International Journal of Computer Vision*, 10:207–231.
- Wesolkowski, S. y Jernigan, E. 1999. "Color edge detection in RGB using jointly Euclidean distance and vector angle". En *Vision Interface'99*, pág. 9–16, Trois-Rivières, Canada.
- Weszka, J. S. 1978. "A survey of threshold selection techniques". *Computer Graphics and Image Processing*, 7:259–265.
- Whatmough, R. J. 1991. "Automatic threshold selection from a histogram using the exponential hull". *CVGIP: Graphical Model and Image Processing*, 53(6):592–600.
- Williams, D. J. y Shah, M. 1990. "Edge contour using multiple scales". *Computer Vision, Graphics and Image Processing*, 51:256–274.
- Witkin, A. P. 1983. "Scale - space filtering". En *Proceedings of 8th Int. Joint Conf. on Artificial Intelligence*, pág. 1019–1022.

- Wright, M. W., Cipolla, R. y Giblin, P. J. 1994. "Skeletonization Using an Extended Euclidean Distance Transform". CUED/F-INFENG/TR 196, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, England.
- Wright, W. A. 1989. "A Markov random field approach to data fusion and colour segmentation". *Image and Vision Computing*, 7(2):144–150.
- Wu, J. y Leou, J. 1993. "New polygonal approximation schemes for object shape representation". *Pattern Recognition*, 26:471–474.
- Wunsch, P. y Laine, A. F. 1995. "Wavelet descriptors for multiresolution recognition of handprinted characters". *Pattern Recognition*, 28(8):1237–1249.
- Wysecki, G. y Stiles, W. S. 1982. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley Series in Pure and Applied Optics. John Wiley & Sons, Inc., segunda edición.
- Xiaohan, Y., Ylä-Jääski, J. y Baozong, Y. 1991. "A new Algorithm for texture segmentation based on edge detection". *Pattern Recognition*, 24(11):1105–1112.
- Xie, M. 1992. "Edge linking by using causal neighborhood". *Pattern Recognition Letters*, 13:647–656.
- Yachida, M. y Tsuji, S. 1970. "Application of colour information to visual perception". *Pattern Recognition*, 3:307–323.
- Yang, C.-K. y Tsai, W.-H. 1996. "Reduction of color space dimensionality by moment-preserving thresholding and its application for edge detection in color images". *Pattern Recognition Letters*, 17(5):481–490.
- Young, I., Walker, J. y Bowie, J. 1974. "An analysis technique for biological shape". *Computer Graphics and Image Processing*, 25:357–370.
- Young, L. y Fu, K. S. 1986. *Handbook of Pattern Recognition and Image Processing*. Academic Press, New York. (Editores).
- Young, R. 1986a. "Simulation of human retinal function with Gaussian derivative model". En *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pág. 564–569, Miami, Florida, E.E.U.U.
- Young, R. 1986b. "The Gaussian derivative model for machine and biological image processing". *General Motors Research Laboratories*, GMR-5128.
- Young, T. 1802. "On the theory of light and colors". *Philophysical Transactions of the Royal Society of London*, 92:20–71.
- Yu, X., Ylä-Jääski, J. y Yuan, B. 1991. "A new algorithm for texture segmentation based on edge detection". *Pattern Recognition*, 24:1105–1112.

- Zahn, C. y Roskies, R. 1972. "Fourier descriptors for plane closed curves". *Computer Graphics and Image Processing*, 21:269–281.
- Zamperoni, P. 1995. "Image enhancement". En *Advances in Imaging and Electron Physics*, pág. 1–76.
- Zenzo, S. D. 1986. "A note on the gradient of a multi-image". *Computer Vision, Graphics and Image Processing*, 33:116–125.
- Zhang, T. Y. y Suen, C. Y. 1984. "A fast parallel algorithm for thinning digital patterns". *Comm. ACM*, 27(3):236–239.
- Zhang, Y. J. 1996. "A survey on evaluation methods for image segmentation". *Pattern Recognition*, 29(8):1335–1346.
- Zhu, P. y Chirlian, P. M. 1995. "On critical point detection of digital shapes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):737–748.
- Zhu, Q. 1996. "Efficient evaluations of edge connectivity and width uniformity". *Image and Vision Computing*, 14:21–34.
- Zhu, S.-Y., Plataniotis, K. N. y Venetsanopoulos, A. N. 1999. "Comprehensive analysis of edge detection in color image processing". *Optical Engineering*, 38(4):612–625.
- Ziou, D. 1991. "Line detection using an optimal IIR filter". *Pattern Recognition*, 24(6):465–478.
- Ziou, D. y Fabre, J. P. 1994. "Effects of edge orientation on the performance of first order operators". *Pattern Recognition Letters*, 15:1053–1063.
- Ziou, D., Fabre, J. P. y Wang, S. 1994. "Performance evaluation of first order operators". En *International Conference on Pattern Letters*, vol. 1, pág. 42–46.
- Ziou, D. y Koukam, A. 1995. "The selection of edge detection using local image structure". En *7th International Conference on Tools with Artificial Intelligence*, pág. 366–370, Virginia, USA.
- Ziou, D. y Tabbone, S. 1993. "A multi-scale edge detector". *Pattern Recognition*, 26(9):1305–1314.
- Ziou, D. y Tabbone, S. 1995. "Adaptative elimination of false edges for first order detectors". En *Lectures Notes on Computer Sciences (Proceedings of the 8th International Conference on Image Analysis and Processing*, vol. 974, pág. 89–94, Italy.
- Ziou, D. y Tabbone, S. 1998. "Edge detection techniques - An overview". *Pattern Recognition and Image Analysis*, 8(4):537–559.

- Ziou, D. y Wang, S. 1996. "Isotropic processing for gradient estimation". En *Proceedings of IEEE, International Conference on Computer Vision and Pattern Recognition*, pág. 660–665, San Francisco.
- Zucker, S. W., Hummel, R. y Rosenfeld, A. 1977. "An application of relaxation labeling to line and curve enhancement". *IEEE Transactions on Computers*, C-26(4):394–403.
- Zugaj, D. y Lattuati, V. 1998. "A new approach of color images segmentation based on fusing region and edge segmentation outputs". *Pattern Recognition Letters*, 31(2):105–113.

Índice de Figuras

1.1	Fases del procesamiento digital de imágenes.	4
1.2	Fases de un algoritmo de reconocimiento de objetos basado en el uso de los puntos críticos de los contornos.	9
2.1	(a) Imagen original; (b) imagen con ruido aditivo gaussiano ($\sigma = 10$).	18
2.2	Filtro paso bajo en el dominio de la frecuencia.	21
2.3	Vecindario de un punto (x_0, y_0)	24
2.4	Máscara de convolución genérica de 3×3 puntos.	24
2.5	Máscaras de convolución del filtro espacial “paso bajo” de (a) 3×3 y (b) 5×5 puntos	25
2.6	Ventana de (a) 5×5 puntos y (b)-(f) subventanas de 3×3 puntos.	26
2.7	Función gaussiana ($\sigma = 1$).	27
2.8	Función gaussiana bidimensional ($\sigma = 1$).	28
2.9	Filtros de suavización: (a) función de Green ($\mu = 1$) y (b) función exponencial simétrica ($a_0 = 0.5$).	29
2.10	Filtro de la mediana usando una máscara de 3×3 puntos.	30
2.11	(a) Imagen original; (b) imagen degradada con un 10% de ruido impulsivo de “sal y pimienta”; (c) imagen obtenida al aplicar el filtro paso bajo; (d) imagen obtenida al aplicar el filtro de la mediana.	31
3.1	(a) Imagen original e (b) información suministrada por un detector de bordes.	37
3.2	Imagen en color con bordes provocados por diferentes motivos.	38

3.3	Tipos de bordes: (a) escalón o salto, (b) rampa convexa, (c) rampa cóncava, (d) escalera, (e) valle, (f) tejado, (g) y (h) pico o pulso.	40
3.4	Borde ideal de tipo escalón o salto: (a) imagen y (b) perfil de la función de intensidad a lo largo de una línea horizontal de la imagen.	40
3.5	Bordes provocados por líneas: (a) imagen y (b) perfil de la función de intensidad a lo largo de una línea horizontal de la imagen.	41
3.6	Imagen con un borde del tipo T-unión.	41
3.7	Bordes provocados por fenómenos físicos.	45
3.8	Perfiles de los modelos de bordes en el espacio de color <i>RGB</i> : (a) <i>albedo</i> - oclusión, (b) sombra - cresta y (c) compuesto - contacto.	46
3.9	Fases de la detección de bordes.	49
3.10	Comportamiento de las derivadas de primer y segundo orden ante la presencia de un borde (González y Woods 1996).	52
3.11	Máscaras de los operadores de (a) Roberts, (b) Prewitt, (c) Sobel, (d) Robinson, (e) Kirsch y (f) del operador isotrópico para calcular la aproximación a G_y	55
3.12	Máscaras del operador laplaciano: (a) conectividad 4; (b) conectividad 8.	56
3.13	(a) Imagen original; (b) imagen obtenida al aplicar el operador laplaciano con conectividad 4.	56
3.14	Laplaciano de la función gaussiana ($\sigma = 1$).	57
3.15	Laplaciano bidimensional de la función gaussiana ($\sigma = 1$).	58
3.16	Defectos en la detección de bordes: (a) borde real, (b) poca robustez frente al ruido, (c) pobre localización y (d) múltiples respuestas.	59
3.17	Comparación entre (a) el operador de Canny y (b) la primera derivada de la función gaussiana ($\sigma = 0.3$).	63
3.18	Filtro de Deriche ($\alpha = \omega = 1$ y $a_1 = -13.3816$).	65
3.19	Filtro de Spacek.	66
3.20	(a) Imagen monocromática, (b) imagen obtenida después de la suavización y diferenciación, (c) histograma de (b), y (d) umbralización con $U = 50$	68

3.21	(a) Imagen generada por un detector de bordes e (b) imagen obtenida al realizar el adelgazamiento de (a) con el algoritmo de Wang-Zhang (1.989) modificado (Krishnapuram y Chen 1993; Wang y Zhang 1989).	71
3.22	Vecindario de tamaño 3×3 utilizado para comprobar si el punto central es un máximo local.	71
3.23	(a) Gradiente de una imagen, (b) gradiente de la misma imagen habiendo suprimido los puntos que no son máximos locales y (c) umbralización de (b).	72
3.24	Fases de la detección de bordes en color mediante reducción de la dimensionalidad.	80
3.25	Fases de la fusión de bordes en una imagen definida en el espacio de color <i>RGB</i> .	82
3.26	Extensión del gradiente monocromático a imágenes definidas en el espacio de color <i>RGB</i> .	84
3.27	Configuraciones para la obtención de los gradientes.	100
3.28	Máscaras para detectar bordes (a) verticales y (b) horizontales.	109
4.1	Tipos de imágenes utilizadas en los estudios de (a y b) Kitchen y Rosenfeld (1.981), (c y d) Owens y Venkatesh (1.989), (e) Spreewers y van der Heijden (1.992), (f) Palmer <i>et al.</i> (1.996) y (g y h) Kanungo <i>et al.</i> (1.995).	117
4.2	Imágenes utilizadas por Ramesh y Haralick (1.992): (a) Einstein y (b) un cerebro.	118
4.3	Estructura del borde de “rampa” utilizada en el estudio de Abdou y Pratt.	124
4.4	Tipos de errores en la detección de bordes: (a) borde ideal, (b) borde fragmentado, (c) borde desplazado y (d) borde engrosado.	125
4.5	Dificultades de la medida de Pratt: (a) imagen con un objeto; (b) y (c) bordes detectados que tienen el mismo valor según dicha medida.	126
4.6	Borde vertical de Venkatesh y Kitchen: (a) imagen sin ruido y (b) nivel de gris de los puntos situados en una línea perpendicular al borde.	128

4.7	Borde orientado de Venkatesh y Kitchen: (a) imagen sin ruido y (b) proporciones utilizadas para calcular el nivel de intensidad de un punto por el que pasa el borde.	129
4.8	Detección de bordes (a) verticales y (b) orientados.	130
4.9	Independencia de los tipos de errores: en cada caso sólo cambia la medida de un tipo de error, permaneciendo los demás iguales. . .	132
4.10	Estructura de una imagen <i>quilt</i>	138
4.11	Máscara de convolución del operador de Laplace.	144
5.1	(a) Contornos de dos objetos y (b-e) pasos de un algoritmo de adelgazamiento de bordes.	155
5.2	Condiciones de los algoritmos de adelgazamiento de bordes: punto P que no puede borrarse debido a que (a) provocaría una discontinuidad o (b) acertaría el extremo de una línea.	156
5.3	Vecindades de orden (a) 4 y (b) 8.	157
5.4	Esqueletos obtenidos por la transformación del eje “medial”: (a) imagen original; (b) imagen con ruido.	158
5.5	Vecindario de un punto P	159
5.6	Ejemplo en el que el algoritmo de Zhang y Suen no genera un esqueleto o contorno unitario.	161
5.7	Casos en los que el algoritmo de Sossa permite borrar el punto central.	162
5.8	Modificación del algoritmo de Wang-Zhang: casos en los que no se debe borrar el punto central.	163
5.9	Esquema de la obtención de la imagen densa.	164
5.10	Máscaras a1 - a10.	165
5.11	Máscaras b1-b9.	166
5.12	Relación entre las máscaras de tipo A y de tipo B.	167
5.13	Restauración de una unión T.	167
5.14	Máscaras a11-a18.	168
5.15	Máscaras a19-a26.	168

5.16	Contorno de un objeto que presenta "espolones" o "picos".	169
5.17	Algoritmo de O'Gorman (1.990): estructura de una ventana $V(x, y, k)$.170	
5.18	Partes de la ventana de 3×3 puntos del algoritmo de O'Gorman.	172
5.19	Ejemplo de un núcleo de tipo ancla para una ventana $V(x, y, 4)$	175
5.20	(a) Silueta de un objeto; (b - g) iteraciones de la versión secuencial del algoritmo de O'Gorman de 3×3 ((O'Gorman 1990) página. 204).	177
5.21	Algoritmo de O'Gorman (1.990): ejemplos de ventanas de 3×3 puntos en dirección (a) norte, (b) sur, (c) este y (d) oeste.	178
5.22	(a) Silueta de un objeto; (b - e) resultados de cada uno de los cuatro subciclos de la primera iteración de la versión paralela del algoritmo de adelgazamiento de O'Gorman de 3×3 ; (f) resultado final obtenido tras 14 subciclos. ((O'Gorman 1990) página 205).	181
6.1	Sistemas de coordenadas: (a) cartesianas, (b) polares respecto del centro de coordenadas; (c) polares respecto del centroide y (d) tangencial.	186
6.2	Direcciones de los códigos de cadena: (a) conectividad 4 y (b) conectividad 8.	187
6.3	Códigos de cadena de (a) 4 direcciones y (b) 8 direcciones.	188
6.4	Contornos y firmas de (a) un círculo y (b) un cuadrado.	189
6.5	(a) Dificultades de la firma: valores múltiples para un mismo ángulo. (b) Envoltorio convexo del contorno (a) (González-Jiménez 2000).	189
6.6	Definiciones alternativas de la firma: (a) distancia al punto opuesto; (b) ángulo de la recta tangente respecto de una recta de referencia.	190
6.7	Concepto del altura de arco (Loncaric 1998).	191
6.8	Fases de una aproximación poligonal de un contorno mediante división.	192

6.9	Aproximación poligonal usando puntos característicos: (a) contorno de la provincia de Córdoba suavizado con un filtro gaussiano ($\sigma = 5$); (b) aproximación poligonal de (a) en la que los vértices son los puntos de curvatura extrema obtenidos con el método de Mokhtarian y Mackworth (1.992).	193
6.10	(a) Envoltente convexa de un contorno. (b) Descomposición del contorno.	194
6.11	Representación mediante tiras: (a) primera descomposición de un contorno y (b) árbol de tiras de (a) (Davis 1986).	195
6.12	Números de Euler de varios objetos: (a) $E = 1$; (b) $E = 0$; (c) $E = -1$	199
7.1	Puntos de discrepancia.	210
7.2	Puntos de discrepancia de dos imágenes arbitrarias.	212
7.3	Variante de la medida de Pratt.	213
7.4	Zona de detección correspondiente a un borde cualquiera.	214
7.5	Independencia de los tipos de errores: en cada caso sólo cambia la medida de un tipo de error, permaneciendo las demás iguales. . .	216
7.6	Imágenes artificiales en color diseñadas: (a) barras 0, (b) barras 30, (c) barras 45, (d) barras 90, (e) círculos, (f) elipses, (g) barras 30 - elipses y (h) barras 0 - barras 90.	219
7.7	Zonas de las imágenes artificiales en color: (a) barras 90 y (b) círculos.	220
7.8	Imágenes “óptimas” obtenidas por las diferentes medidas de calidad, al umbralizar la imagen generada al aplicar el detector de Di Zenzo a la imagen barras 90 sin ruido. El porcentaje indica el umbral donde cada medida obtiene el valor óptimo.	223
7.9	Curvas de error de la medidas de calidad correspondientes a la imagen generada por el detector de Di Zenzo al ser aplicado a la imagen barras 90 sin ruido.	224
7.10	Imágenes “óptimas” obtenidas por las diferentes medidas de calidad, al umbralizar la imagen generada al aplicar el detector de Scharcanski a la imagen elipses con ruido gaussiano ($\sigma^2 = 3.0$). El porcentaje indica el umbral donde cada medida obtiene el valor óptimo.	225

7.11	Curvas de error de las medidas de calidad correspondientes a la imagen generada por el detector de Scharcanski al ser aplicado a la imagen elipses con ruido gaussiano ($\sigma^2 = 3.0$).	226
7.12	Imágenes “óptimas” obtenidas por las diferentes medidas de calidad al umbralizar la imagen generada al aplicar el detector Dv - hv a la imagen barras 90 con ruido exponencial ($\sigma^2 = 3.0$). El porcentaje indica el umbral donde se obtiene el valor óptimo. . . .	227
8.1	Resultados obtenidos por los detectores de bordes al ser aplicados a la imagen barras 30 - elipses sin ruido. En cada caso se muestra la imagen umbralizada y el valor de la medida de Baddeley en la que cada detector alcanza el valor óptimo.	232
8.2	Valores medios obtenidos por los detectores de bordes en color al ser aplicados a las imágenes artificiales con diferentes niveles de ruido gaussiano o exponencial.	233
8.3	Resultados obtenidos por los detectores de bordes al ser aplicados a la imagen barras 0 - barras 90 con ruido exponencial de varianza $\sigma^2 = 1.0$. En cada caso se muestra la imagen umbralizada y el valor de la medida de Baddeley en la que cada detector alcanza el valor óptimo.	235
8.4	Imágenes de consenso obtenidas al utilizar un umbral del 20% en las imágenes generadas por los detectores de bordes al ser aplicados a la imagen en color Lenna (figura 8.5). La imagen de nivel de consenso i está compuesta por aquellos puntos que han sido considerados como puntos de borde por “al menos” i detectores. .	236
8.5	Imágenes obtenidas al utilizar un umbral del 20% en las imágenes generadas por los detectores de bordes al ser aplicados a la imagen en color Lenna.	237
8.6	Valores medios obtenidos por los detectores de bordes en color al ser aplicados a las imágenes reales y utilizar un umbral del 20%. .	238
9.1	Criterios del algoritmo de adelgazamiento con conectividad 8: (a) patrón que cumple los criterios; (b) patrón con puntos marcados del perímetro que no están conectados entre sí mediante conectividad 8; (c) patrón en el que el punto central tiene menos de dos vecinos marcados; (d) patrón en el que la cadena más larga de puntos no marcados del perímetro no es superior a 1.	244
9.2	Máscara de convolución utilizada por el algoritmo de adelgazamiento.	244

9.3	Codificación numérica de los patrones que permiten eliminar puntos del lado norte de un contorno respetando la conectividad de orden 8.	245
9.4	Tipos de patrones.	245
9.5	Fases del algoritmo de adelgazamiento de bordes con conectividad 8: (a) contorno original; eliminación de puntos de los lados (b) norte (primera vez), (c) sur, (d) este, (e) oeste y (f) norte (segunda vez); (g) contorno adelgazado.	246
9.6	Máscaras de convolución utilizadas al eliminar los puntos de la capa (a) este, (b) sur y (c) oeste.	247
9.7	Codificación numérica de los patrones que permiten eliminar puntos del lado norte de un contorno respetando la conectividad de orden 4.	248
9.8	Esqueletos obtenidos por los algoritmos basados en (a) la conectividad de orden 8 y (b) de orden 4; (c) diferencias entre ambas versiones.	249
9.9	Adelgazamiento de un contorno: (a) contorno de una albacora; (b) puntos borrados por el algoritmo basado en la conectividad de orden 8.	249
10.1	Atún: (a) imagen en color y (b) contorno exterior.	252
10.2	Identificación de los puntos del contorno exterior de un objeto mediante el método del sondeo.	253
10.3	Dificultades del método del sondeo.	254
10.4	Identificación de los puntos del contorno exterior de un objeto mediante el método de la pila.	255
10.5	Justificación del uso de la conectividad 4 al identificar los puntos del contorno exterior con el método de la pila: (a) imagen original; puntos obtenidos utilizando (b) conectividad 4 y (c) conectividad 8.	256
10.6	Dificultades del método de la pila.	257
10.7	Funcionamiento de los métodos de identificación de los puntos del contorno exterior cuando éste está abierto: (a) imagen original; puntos del contorno exterior identificados por los métodos del sondeo (b) y de la pila (c).	258

10.8	Funcionamiento de los métodos de identificación de los puntos del contorno exterior cuando éste está cerrado: (a) imagen original; puntos del contorno exterior identificados por los métodos del sondeo (b) y de la pila (c).	259
10.9	Recorridos de los contornos obtenidos por el método de extracción: (a) y (b) contornos cerrados; (c) contorno abierto; (d) dos contornos.	261
10.10	Eliminación de contornos con pocos puntos: (a) imagen con contornos con puntos ruidosos; (b) imagen obtenida al eliminar los contornos con menos de 20 puntos.	262
10.11	Dificultades del método de extracción: (a) contorno con “picos” y “cruce”; (b) contornos extraídos.	263
10.12	Aplicación del algoritmo de supresión de “picos”: (a) imagen original; (b) primera iteración y (c) segunda iteración.	263
10.13	Mejora del algoritmo de supresión de “picos”: (a) imagen original; (b) adelgazamiento de (a) con un algoritmo con conectividad 8 y (c) supresión de “picos” de (b).	264
10.14	Ejemplos de contornos cerrados: (a) puntos extremos y recorrido de un contorno cerrado; (b) contorno exterior cerrado de una bombilla.	265
10.15	Cierre de un contorno abierto: (a) recorrido de un contorno abierto; (b) cierre del contorno (a).	266
10.16	Cerramiento de contornos abiertos de una bombilla: (a) contorno con una apertura superior; (b) cierre del contorno (a); (c) contorno con una apertura lateral; (d) cierre del contorno (c).	267
10.17	Pasos de la unión de tres contornos.	268
10.18	Unión de varios contornos: (a) contornos de una bombilla; (b) cierre del contorno (a).	269
10.19	Mejora del método de unión de varios contornos: (a) contornos de un objeto; (b) unión de los contornos de (a); (b) adelgazamiento con conectividad 8 de (b); (c) supresión de “picos” de (c).	269
10.20	Dificultades en la unión de contornos: (a) contornos de una imagen; (b) cruce provocado por el algoritmo de unión.	270
10.21	Dificultades del algoritmo de unión de contornos: (a) imagen con varios contornos; (b) unión de los contornos de la imagen (a) que provoca un cruce.	270

11.1	Dificultades del método básico de parametrización de un contorno: (a) recorrido del contorno original; (b) recorrido del contorno original rotado 30°	272
11.2	Eje de mínima inercia.	273
11.3	Eje de mínima inercia que no intersecta al contorno.	273
11.4	Distancia de los puntos del contorno al eje de mínima inercia. . .	274
11.5	Parametrizaciones de un contorno: (a) eje de mínima inercia; (b) primera parametrización; (c) segunda parametrización.	274
11.6	Aplicación del algoritmo de Mokhtarian y Mackworth: (a) contorno original; (b) contorno suavizado con el filtro gaussiano ($\sigma = 5$); (c) puntos de curvatura cero sobre el contorno suavizado; (d) aproximación lineal del contorno a partir de los puntos de curvatura cero; (e) puntos de curvatura extrema sobre el contorno suavizado; (f) aproximación lineal del contorno a partir de los puntos de curvatura extrema.	277
11.7	Obtención de los puntos críticos de dos contornos ($\sigma = 5$): (a) contorno original 1; (b) contorno original 2; (c) puntos de curvatura extrema sobre el contorno suavizado 1; (d) puntos de curvatura extrema sobre el contorno suavizado 2; (e) puntos de curvatura extrema y equivalentes sobre el contorno suavizado 1; (f) puntos de curvatura extrema y equivalentes sobre el contorno suavizado 2.	279
12.1	Ejemplo de imagen en color utilizada en la primera prueba de comparación: (a) besugo; (b) contorno de (a).	292
12.2	Dos de las imágenes en color utilizadas en la segunda prueba de comparación de contornos: (a) arenque 1; (b) arenque 2.	300
12.3	Contornos exteriores de los peces de la figura 12.2: (a) contorno de arenque 1; (b) contorno de arenque 2.	301
12.4	Media de los valores de la función basada en el ajuste de los ángulos obtenidos por cada pez para diferentes escalas.	305
12.5	Media de los valores de la función basada en el ajuste de los ángulos obtenidos por cada pez para diferentes orientaciones.	305
12.6	Imágenes reales en color de peces utilizados en la tercera prueba: (a) caballa; (b) dorada; (c) lubina.	306
12.7	Contornos extraídos de las imágenes reales de la figura 12.6: (a) caballa; (b) dorada; (c) lubina.	308

12.8	Media de los valores de la función de ajuste de los ángulos para cada una de las especies.	310
12.9	Media de los valores de la función de ajuste de los ángulos para cada uno de los peces.	313
A.1	Espectro electromagnético.	321
A.2	Sensibilidad de los conos de la retina.	323
A.3	Espacio de color <i>RGB</i>	326
A.4	Diagrama cromático del sistema XYZ.	330
A.5	Espacio de color <i>HSI</i>	334
A.6	Espacios de color: (a) <i>HLS</i> y (b) <i>HSV</i>	335
A.7	Espacio de color <i>TekHVC</i>	335
A.8	Imagen en color y sus planos de color.	338
A.9	Estructura de una imagen en color.	339
A.10	Punto o <i>pixel</i> de una imagen en color.	340

Índice de Tablas

3.1	Clasificación de los bordes propuesta en (Stokman y Gevers 1999) ("+ denota sensibilidad; "-", invarianza).	42
3.2	Clasificación de los bordes propuesta en (Stokman y Gevers 2000a) ("+ denota sensibilidad; "-", invarianza).	43
3.3	Clasificación de los bordes propuesta en (Stokman y Gevers 2000b) ("+ denota sensibilidad; "-", invarianza).	44
4.1	Coeficientes de las medidas de los tipos de error.	133
4.2	Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (primera parte).	148
4.3	Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (segunda parte).	149
4.4	Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (tercera parte).	150
4.5	Estudios comparativos de algoritmos de detección de bordes en imágenes monocromáticas (cuarta parte).	151
4.6	Estudios comparativos de algoritmos de detección de bordes en imágenes en color (primera parte).	152
4.7	Estudios comparativos de algoritmos de detección de bordes en imágenes en color (segunda parte).	153
4.8	Estudios comparativos de algoritmos de detección de bordes en imágenes en color (tercera parte).	154
7.1	Coeficientes de las medidas de los tipos de error utilizados en la variante general de Venkatesh y Kitchen.	217

7.2	Valores de las componentes de color de cada una de las zonas de las imágenes artificiales.	221
8.1	Rendimiento de los detectores de bordes en color con imágenes artificiales: subconjuntos homogéneos según el test de Scheffé. Se muestran las medias para los grupos en subconjuntos homogéneos.	234
8.2	Imágenes de consenso: subconjuntos homogéneos según el test de Scheffé. La tabla muestra las medias para los grupos en subconjuntos homogéneos teniendo en cuenta todos los umbrales porcentuales utilizados (10%, 15% y 20%).	239
8.3	Rendimiento de los detectores de bordes en color con imágenes reales utilizando sólo imágenes de consenso comprendidas entre 3 y 7 y todos los umbrales porcentuales utilizados (10%, 15% y 20%). Se muestran los valores medios de los subconjuntos homogéneos según el test de Scheffé.	240
12.1	Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación de ajuste de los ángulos (primera parte).	296
12.2	Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación de ajuste de los ángulos (segunda parte).	297
12.3	Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación basada en el ajuste de los ángulos y los radios de los puntos críticos (primera parte).	298
12.4	Resultados obtenidos al comparar un besugo con los demás peces utilizando la función de comparación basada en el ajuste de ángulos y los radios de los puntos críticos (segunda parte).	299
12.5	Resultados obtenidos al comparar el arenque 1 rotado 30° y escalado 0.5 con los demás peces utilizando la función de comparación de ajuste de los ángulos.	302
12.6	Tabla ANOVA de los resultados de la función basada en el ajuste de los ángulos al comparar arenque con distintas orientaciones y escalas.	303
12.7	Subconjuntos homogéneos para los valores de Grados 1 obtenidos mediante el test de Scheffé: se muestran las medias de los grupos en los subconjuntos homogéneos.	304

12.8 Subconjuntos homogéneos para los valores de Escala 1 obtenidos mediante el test de Scheffé: se muestran las medias de los grupos en los subconjuntos homogéneos. 304

12.9 Descriptores estadísticos de la función de ajuste de los ángulos para cada una de las especies. 309

12.10 Descriptores estadísticos de la función de ajuste de los ángulos para cada uno de los peces. 311

12.11 Frecuencias de las iteraciones necesarias para el alcanzar el valor mínimo durante las pruebas con imágenes reales usando la función de ajuste de los ángulos. 312

A.1 Sensibilidades máximas de los tipos de conos de la retina. 323

