

Introducción al Procesamiento y Análisis de Imágenes Digitales¹

R. MOLINA

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada. 18071 Granada,

Teléfono: +34 58 243 197, Fax: + 34 58 243317,

¹Curso impartido en Introducción a la Robótica hasta 1998

Índice general

1. Introducción	23
1.1. ¿Qué es la visión artificial?	23
1.2. Nuestro Patrocinador: El Ojo	26
1.2.1. La retina	28
1.2.2. Los caminos visuales al cerebro	29
1.3. Ilusiones, Ambigüedades e Inconsistencias	31
2. Fundamentos	37
2.1. Elementos de un sistema de procesamiento de imágenes	37
2.2. Modelo de Imagen Simple	38
2.3. Geometría de Imágenes	40
2.3.1. Perspectiva por Proyección	40
2.3.2. Proyección Ortográfica	41
2.4. Efectos del muestreo y la cuantificación	42
2.4.1. Efectos del muestreo	42
2.4.2. Efecto de la cuantificación	42
2.5. Formación de la Imagen	44
2.5.1. Función de Emborronamiento	45
2.5.2. Transformación	48
2.5.3. Modelos de Ruido	49
2.6. Algunas Relaciones Básicas Entre Píxeles	49
2.6.1. Vecinos de un píxel	49
2.6.2. Conectividad	50
2.6.3. Medidas de Distancia	51
2.6.4. Operaciones Aritmético / Lógicas	52

2.7. Pasos Fundamentales en el Procesamiento de Imágenes	53
2.8. Ejercicios	56
3. Sobre las imágenes y su almacenamiento en ficheros	59
3.1. Introducción	59
3.2. Fundamentos del Color	59
3.3. Modelos de Color	61
3.3.1. El modelo de color RGB	62
3.3.2. El modelo de color CMY	65
3.3.3. El modelo de color YUV	65
3.3.4. El modelo de color YIQ	66
3.3.5. El modelo de color HSI	66
3.4. Formatos de ficheros de imágenes	68
3.5. Ejercicios	69
4. Transformaciones de Imágenes	71
4.1. Transformada de Fourier. Caso Continuo.	71
4.1.1. Transformada de Fourier Unidimensional	71
4.1.2. Interpretación física de la transformada de Fourier unidimensional	75
4.1.3. Transformada de Fourier Bidimensional	75
4.1.4. Interpretación física de la transformada de Fourier bidimensional	77
4.1.5. Propiedades de la Transformada de Fourier bidimensional	78
4.1.6. Convolución y Correlación	78
4.2. Señal continua y muestreo discreto. Reconstrucción de la señal continua a partir del muestreo	81
4.3. Series de Fourier.	86
4.4. Recuperación de la señal original a partir del muestreo en un número finito de puntos	87
4.5. Desde la transformada de Fourier Continua hacia la discreta	89
4.6. Transformada de Fourier Discreta. Caso Bidimensional.	90
4.6.1. Convolución y Transformada de Fourier	92
4.6.2. Correlación	94
4.7. Algunos ejemplos de Transformadas de Fourier	95
4.8. Uso de la transformada de Fourier para resolver sistemas de ecuaciones	95

4.8.1. Fourier, Convolución y Matrices Circulantes por Bloques	95
4.8.2. Propiedades del cálculo matricial usando coeficientes de Fourier . . .	98
4.8.3. Resolución de Ecuaciones Lineales Mediante Transformada de Fourier	99
4.8.4. Transformada de Fourier Rápida. Implementación y aplicaciones en procesamiento de imágenes	99
4.9. Ejercicios	105
5. Restauración de Imágenes	107
5.1. Restauración sin condiciones, Filtrado Inverso	107
5.1.1. Restauración con restricciones	108
5.2. Filtro de Wiener	109
5.3. Ejercicios	111
6. Mejora de Imágenes	113
6.1. Introducción	113
6.2. Mejora Mediante Procesamiento Puntual	113
6.2.1. Ampliación de Contraste	113
6.2.2. Corte y Umbralización	114
6.2.3. Negativo Digital	115
6.2.4. Extracción de Bits	116
6.2.5. Compresión del Rango	116
6.2.6. Potenciación	116
6.2.7. Diferencia de Imágenes	117
6.2.8. Promedio de Imágenes	117
6.3. Modificación del Histograma	117
6.3.1. Uniformización del Histograma	118
6.3.2. Especificación del Histograma	121
6.3.3. Mejora Local	123
6.4. Filtrado Espacial	124
6.4.1. Filtros de Alisamiento	125
6.4.2. Filtros de Realce	127
6.5. Mejora en el Dominio de las Frecuencias	129
6.5.1. Filtrado de Paso Bajo	131
6.5.2. Filtrado de Paso Alto	133

6.6. Ejercicios	133
7. Segmentación Basada en Fronteras	137
7.1. Métodos de Gradiente	140
7.1.1. Operadores Isotrópicos	141
7.2. Pasos en la detección de Bordes	142
7.3. Aproximaciones Discretas del Gradiente	142
7.3.1. Operador de Roberts	143
7.3.2. Operador de Sobel	143
7.3.3. Operador de Prewitt	144
7.3.4. Otros Operadores de Derivadas	144
7.4. Operadores de Segunda Derivada	144
7.4.1. Operador Laplaciano	144
7.4.2. Cruces Por cero de la segunda Derivada. Laplaciana de Gaussiana	146
7.5. Detectores de Bordes Gaussianos	149
7.5.1. El detector de Bordes de Canny	149
7.6. Medidas de la Calidad de un detector de bordes	151
7.7. Técnicas Jerárquicas	151
7.8. Técnicas de Segmentación Basadas en Bordes	154
7.8.1. Relajación de Bordes	154
7.9. Ejercicios	158
8. Segmentación de Imágenes	159
8.1. Introducción	159
8.2. Regiones y Bordes	160
8.3. Segmentación Basada en Umbralización	161
8.3.1. Método P-cuantil	162
8.3.2. Método de la Moda	164
8.3.3. Umbralización Adaptativa	165
8.3.4. Umbralización en Estructuras de Datos Jerárquicas	166
8.3.5. Selección del Umbral Basada en Características de la Frontera	167
8.4. Técnicas de Segmentación Basadas en Bordes	167
8.4.1. Extracción de la Frontera	168
8.4.2. Transformada de Hough	171

8.5. Segmentación Orientada a Regiones	172
8.5.1. Unión de Regiones	172
8.5.2. Unión de regiones por agregación de píxeles	173
8.5.3. Separación y Unión de Regiones	174
8.6. Ejercicios	176
9. Transformada de Hough	179
9.1. Motivación.	179
9.1.1. Problema.	179
9.1.2. Soluciones simples.	180
9.1.3. Objetivo.	181
9.2. Transformada de Hough.	182
9.2.1. Detección de líneas.	183
9.2.2. Ejemplos.	192
9.2.3. Generalización a una curva cualquiera.	193
9.2.4. Ejemplos.	202
9.3. Eficiencia de la Transformada de Hough	202
9.3.1. Correlación vs. Transformada de Hough	202
9.3.2. Requisitos de espacio y tiempo.	203
9.3.3. Mejora de eficiencia: Uso del gradiente.	205
9.4. Ejercicios.	209
9.4.1. Transformada de Hough para localizar líneas rectas.	209
9.4.2. Uso del módulo del gradiente.	210
9.4.3. Detección de segmentos.	210
9.4.4. Unión de fronteras.	210
9.4.5. Uso del gradiente en la detección de una curva cualquiera con rota- ciones.	211
10. Representación y Descripción de Fronteras y Regiones	221
10.1. Introducción	221
10.2. Representaciones de la Frontera	222
10.2.1. Cadenas de códigos	222
10.2.2. Aproximaciones Poligonales	222
10.2.3. Signatura	223

10.3. Representaciones de las regiones	226
10.3.1. Códigos de Longitud Variable	226
10.3.2. Quad-trees	226
10.3.3. Proyecciones	226
10.4. Descriptores de Fronteras	230
10.4.1. Algunos descriptores simples	230
10.4.2. Descriptores de Fourier	230
10.4.3. Momentos	233
10.5. Descriptores de Regiones	233
10.5.1. Algunos descriptores simples	234
10.5.2. Momentos	234
10.5.3. Texturas	235
10.6. Ejercicios	237
11. Introducción a la Morfología Matemática	239
11.1. Introducción	239
11.2. Operaciones básicas sobre conjuntos	241
11.3. Morfología Binaria	241
11.3.1. Dilatación Binaria	241
11.3.2. Erosión binaria	243
11.3.3. Algoritmos morfológicos basados en la erosión y la dilatación	246
11.3.4. Transformación Ganancia-Pérdida	248
11.3.5. Algoritmos de morfología basados en la transformación Ganancia-Pérdida	251
11.3.6. Apertura y Clausura	254
11.3.7. Algoritmos morfológicos basados en aperturas y clausuras	257
11.4. Morfología de niveles de gris	258
11.4.1. Dilatación	258
11.4.2. Erosión	260
11.4.3. Apertura y Clausura	260
11.4.4. Algunas aplicaciones de la morfología de niveles de gris	262
12. Introducción a la Compresión de Imágenes	265
12.1. Introducción	265

12.1.1. Ratios de Información	266
12.2. Cuantificación	266
12.2.1. Cuantificador del óptimo cuadrado medio o de Lloyd-Max	267
12.2.2. El cuantificador uniforme optimal	267
12.3. Codificación de pixels	268
12.3.1. Modulación de códigos de pulso o PCM	268
12.3.2. Codificación por entropía	268
12.3.3. Codificación por longitud de recorridos (Run-Lenght Coding)	270
12.3.4. Codificación de planos de bits	271
12.4. Técnicas predictivas	271
12.4.1. Modulación Delta	272
12.4.2. DPCM línea a línea	272
12.4.3. DPCM bidimensional	273
12.4.4. Técnicas Adaptativas	273
12.5. Teoría de codificación por transformación	275
12.5.1. Codificador por transformación óptimo	275
12.5.2. Codificación de imágenes por transformación	276
12.6. Codificación híbrida y DPCM de vector	277
12.6.1. Idea básica	277
12.6.2. Algoritmo de codificación híbrida	277
12.6.3. Codificación híbrida adaptativa	278
12.7. Codificación de imágenes en color y multiespectrales	278
12.8. El estándar de compresión de imágenes estáticas JPEG	280
12.8.1. Introducción	280
12.8.2. Pasos del proceso para codificación basado en la DCT	281
12.8.3. Modelo Base y otros codecs secuenciales basados en la DCT	288
A. Modelos de Visión	289
A.1. El trabajo de David Marr	290
A.1.1. Niveles del Análisis de Escenas	291
A.2. El Esbozo Primitivo	293
A.3. Hacia el esbozo $2\frac{1}{2} D$. Percepción de la profundidad y el movimiento	304
A.3.1. Percepción Estereoscópica	304
A.3.2. Percepción del Movimiento	309

A.3.3.	Sobre la percepción de otros aspectos de la visión	314
A.3.4.	Recapitulación sobre la teoría del esbozo $2\frac{1}{2}D$ de Marr	314
A.4.	Teoría del Reconocimiento del Objeto de Marr y Nishihara	318
B.	De la Escena 3-D a la Imagen 2-D	325
B.1.	Introducción	325
B.2.	Formación de Imágenes	325
B.2.1.	Perspectiva por Proyección	326
B.3.	Introducción a la Fotometría	327
B.3.1.	Intensidad luminosa de una fuente puntual (I)	327
B.3.2.	Ley del cuadrado de la distancia. Relación Intensidad-Illuminancia	328
B.3.3.	Fuentes no Puntuales. Luminancia	329
B.3.4.	Emisores y Difusores perfectos. Ley de Lambert	330
B.3.5.	Ley de lambert	331
B.3.6.	Relaciones Fotométricas	332
C.	Preliminares Matemáticos	337
C.1.	Resultados de Teoría de Matrices	337
C.1.1.	Vectores y Matrices	337
C.1.2.	Ordenación por Filas y por Columnas	339
C.1.3.	Reglas de Trasposición y Conjugación	339
C.1.4.	Matrices Circulantes y de Toeplitz	340
C.1.5.	Matrices Ortogonales y Unitarias	340
C.1.6.	Matrices Definidas Positivas y Formas Cuadráticas	341
C.1.7.	Formas Diagonales	341
C.2.	Matrices por Bloques y Matrices de Kronecker	342
C.2.1.	Matrices por Bloques	342
C.2.2.	Productos de Kronecker	342
C.2.3.	Operaciones Separables	343
C.3.	Señales Aleatorias	344
C.3.1.	Definiciones	344
C.3.2.	Distribución Normal o Gaussiana	345
C.3.3.	Procesos Aleatorios Gaussianos	345
C.3.4.	Procesos Estacionarios	345

C.3.5. Procesos de Markov	346
C.3.6. Ortogonalidad e Independencia	346
C.4. Campos Aleatorios Discretos	347
C.5. La Función de Densidad Espectral	348

Índice de figuras

1.1. Sistema de visión de propósito general	26
1.2. La espiral de Fraser. Ilusión de espirales	27
1.3. El ojo humano	29
1.4. La retina	30
1.5. Distribución de conos y bastones	30
1.6. Caminos visuales del ojo al cerebro	32
1.7. Seis ilusiones ópticas clásicas. En la ilusión de Zöllner las diagonales son paralelas pero no lo parecen, en la de Poggendorff, la línea recta no lo parece, en la de Helmholtz los cuadrados parecen rectangulares, en la de Müller-Lyer las líneas horizontales parecen de diferente longitud y en los de Hering y Wundt las líneas rectas parecen curvadas.	33
1.8. Este dibujo indio es ambiguo, tenemos a Poseidón y una sirena	34
1.9. Cuatro ejemplos clásicos de ambigüedad	34
1.10. Consistencias Geométricas	35
2.1. Un Mondrian formado por regiones de color constante	38
2.2. De escena 3-D a la imagen 2-D	40
2.3. De izquierda a derecha y de arriba a abajo. Muestreo a 256×256 , 128×128 , 64×64 y 32×32 píxeles	43
2.4. Efecto de la cuantificación. De izquierda a derecha, 8 y 1 bits.	44
2.5. Captación de Imagen	45
2.6. Imagen original y borrosa por desenfoque.	47
2.7. Imagen original y ruidosa con ruido gaussiano.	50
2.8. Pasos en el procesamiento de la imagen	56
3.1. suma de colores primarios	61

3.2.	Colores secundarios	62
3.3.	Cubo de color RGB. Los puntos en la diagonal principal tienen niveles de gris desde el negro en el origen al blanco en el punto $(1, 1, 1)$	63
3.4.	(a) Imagen original en color. (b) Plano de rojo. Plano de verde y (c) plano de azul	64
3.5.	(a) Plano C del modelo CMY (b) Plano M del modelo CMY (c) Plano Y del modelo CMY (d) Plano Y del modelo YIQ (e) Plano I del modelo YIQ (f) Plano Q del modelo YIQ.	67
4.1.	Representación de $\cos(2\pi x)$, línea continua, y $\cos(2\pi 2x)$, línea discontinua que corresponden a uno o dos ciclos por unidad de tiempo.	73
4.2.	Función $\frac{\text{sen}(\pi 10x)}{\pi 10x}$	74
4.3.	Líneas correspondientes a máximos de la función $\cos(2\pi(ux + vy))$	77
4.4.	Convolución Unidimensional.	80
4.5.	Ejemplo de convolución.	81
4.6.	Representación gráfica de la convolución en dos dimensiones. Las dos funciones a convolucionar aparecen en (a) y (b). En (c) la primera función se rota 180° y (d) muestra el producto $g(x - \alpha, y - \beta)f(\alpha, \beta)$. El volumen bajo la función producto es la evaluación de la convolución para el valor de (x, y) elegido.	82
4.7.	Muestreo	84
4.8.	Muestreo de duración finita	88
4.9.	De izquierda a derecha y de arriba a abajo. Imagen y su espectro de potencias, la misma imagen rotada y su espectro de potencias	96
5.1.	Imagen observada de Saturno a la izquierda y restaurada a la derecha	110
6.1.	Ampliación del Contraste	114
6.2.	Imagen original y su umbralización por 150	115
6.3.	Ejemplos de histograma	118
6.4.	Una transformación de los niveles	119
6.5.	Ilustración de transformación uniforme	122
6.6.	Ejemplo de igualación del histograma	122
6.7.	Imagen y su filtro de paso alto.	128
6.8.	Imagen y su realce al	129

6.9.	(a) Imagen original, (b) Butterworth de orden 1 y corte 0,5. (c) Butterworth de orden 6 y corte 0,5.(d) Butterworth de orden 1 y corte 0,1.	134
7.1.	Tipos de Discontinuidades en una escena. Ver texto	138
7.2.	Perfiles de bordes. Modelos idealizados, fila superior y modelos observados, fila inferior.	139
7.3.	Etiquetado de niveles de gris para explicar los detectores de bordes	144
7.4.	Salto y Resultado de Aplicar el Laplaciano	146
7.5.	Otro Salto y Resultado de Aplicar el Laplaciano	147
7.6.	(a) Intensidad, (b) Derivada Primera (c) Derivada segunda	147
7.7.	Defectos de Bordes	152
7.8.	Borde tipo grieta e con sus vecinos	155
7.9.	Borde tipo grieta e con sus vecinos	157
8.1.	Imagen original y su histograma.	163
8.2.	Problemas con la conectividad.	168
8.3.	Ejemplo de Seguimiento de vontorno.	169
8.4.	Unión heurística de bordes	170
8.5.	(a) Línea recta y (b) su transformada de Hough	171
8.6.	Ejemplo de crecimiento de regiones a partir de puntos semilla.	173
8.7.	Ejemplo de quad-tree	175
8.8.	Proceso de separación y unión usando quad-trees.	177
9.1.	Motivación de la transformada de Hough.	182
9.2.	Dualidad punto-línea.	184
9.3.	Discretización del espacio m-c.	186
9.4.	Transformada de Hough usando el espacio m-c.	187
9.5.	Representación “normal” de la recta y discretización del espacio θ, ρ	191
9.6.	Transformada de Hough usando el espacio $\theta - \rho$	212
9.7.	Ejemplo sencillo de 4 líneas rectas.	213
9.8.	Detección de líneas de una imagen real.	214
9.9.	Detección de líneas en una imagen ruidosa.	215
9.10.	Transformada de Hough para la localización de círculos.	216
9.11.	Parametrización de una elipse.	217
9.12.	Transformada de Hough para la localización de una curva cualquiera. . . .	218

9.13. Ejemplos de generalización.	219
9.14. Geometría de la R-Tabla en la Transformada de Hough Generalizada.	220
9.15. Errores en el gradiente.	220
10.1. Direcciones utilizadas para codificar una frontera	222
10.2. Codificaciones de una frontera. (a) Frontera, (b) 4-conectividad, (c) 8- conectividad.	223
10.3. Representando una frontera por aproximaciones poligonales.	224
10.4. Signatura de dos fronteras distintas.	225
10.5. Codificaciones de una región. (a) Imagen binaria y su codificación.	227
10.6. (a) Región, (b) Quad-tree y su representación	228
10.7. Una imagen y dos de sus proyecciones.	229
10.8. Frontera digital y su representación compleja	231
10.9.	233
10.10(a) Segmento de Frontera, (b) Representación 1-D	234
10.11Algunos ejemplos de texturas	235
11.1. Ejemplo de dilatación. (a) Elemento estructural, B . (b) Imagen, A . (c) Resultado de la dilatación $A \oplus B$	242
11.2. Otro ejemplo de dilatación. (a) Elemento estructural, B . (b) Imagen, A . (c) Resultado de la dilatación $A \oplus B$	243
11.3. Ejemplo de erosión. (a) Elemento estructural, B . (b) Imagen, A . (c) Resul- tado de la erosión $A \ominus B$	244
11.4. Extracción de fronteras usando la erosión	247
11.5. Relleno de regiones	249
11.6. A contiene una componente conexa que comienza en p , (tono gris para distinguirlo de los otros. En (c) el resultado de la primera aplicación del proceso iterativo, (d) el segundo paso (e) el tercero y (e) el último	250
11.7. Aplicación de la transformación ganancia-pérdida para encontrar esquinas superiores derechas	251
11.8. Cálculo de la envolvente convexa de un conjunto	252
11.9. Elementos estructurales para el cálculo de la m -conectividad	253
11.10Extracción de cuerpo y asa. (a) contiene la imagen A , (b) el elemento estructural K , (c) la apertura $A \circ K$ y (d) $A - A \circ K$	256

11.11(a) Elemento estructural b , (b) imagen f , (c) $f \oplus b$. (d) relleno de b , notado $U(b)$, (e) relleno de f , notado $U(f)$, la extracción de la parte superior de $U(f) \oplus U(b)$ que se nota $T(U(f) \oplus U(k))$ produce $f \oplus b$	259
11.12(a) Elemento estructural b , (b) imagen f , (c) $f \ominus b$. (d) relleno de b , notado $U(b)$, (e) relleno de f , notado $U(f)$, la extracción de la parte superior de $U(f) \ominus U(b)$ que se nota $T(U(f) \ominus U(k))$ produce $f \ominus b$	261
11.13(a) Elemento estructural que se interpreta como esfera, (b) imagen, (c) proceso para obtener la apertura, (d) proceso para obtener la clausura . . .	262
12.1. Particionamiento del intervalo $[0,1]$ de la recta real en subintervalos	270
12.2. Salida típica para un codificador predictivo DM	272
12.3. Pixels A , B , C y D usados en predicción bidimensional	273
12.4. Codificación de componentes de imágenes en color.	279
12.5. Pasos del proceso de codificación basado en la DCT	281
12.6. Pasos del proceso de decodificación basado en la DCT	282
12.7. Codificación diferencial de los coeficientes DC	285
12.8. Secuencia zig-zag para la preparación de los coeficientes cuantificados para la codificación por entropía	285
A.1. Imagen original (arriba) hecha borrosa por filtros gaussianos de dos amplitudes diferentes (abajo)	295
A.2. (a) Intensidad, (b) Derivada Primera (c) Derivada segunda	296
A.3. Resultado de pasar una imagen (superior izquierda) a través de filtros $\nabla^2 G$ con tres amplitudes diferentes, estrecho (superior derecho), intermedio (inferior derecho) y ancho (inferior derecho)	296
A.4. en (a) y (b) se muestran los pasos por cero de la imagen en figura con un filtro estrecho y otro ancho, cuando se combinan ambas, las localizaciones de las manchas, los segmentos de borde y las barras se muestran en (c), (d) y (e)	297
A.5. Agrupación de puntos. a, b y c se agrupan en A	299
A.6. Agrupación curvilínea. a, b, c, d y sucesivos se agrupan en A	299
A.7. Ejemplo de agregación theta	300
A.8. Agrupación curvilínea junto con clausura puede revelar el contorno a-b-c-d	301
A.9. Osito de peluche, esbozo primario en bruto y agrupamiento	302

A.10.(a) Hoja, (b) esbozo primario en bruto, (c) y (d) algunas agrupaciones curvilineas, (e) la separación no es posible (f) y (g) estructuras separadas tras decirle que los elementos de 1 y 2 no van juntos	303
A.11.Los ojos enfocan a A y se proyectan en los puntos de las retinas a_1 y a_2 . B se proyecta en dos puntos más dispares	305
A.12.Estereoscopio de Wheatstone	306
A.13.Estereograma de puntos aleatorios. Si este par estereo fuese visto en un estereoscopio se vería un cuadrado flotando por encima del fondo	307
A.14.(a) y (b) son las dos mitades de un estereograma de puntos aleatorios simplificado. Con una zona circundante común (S) y una zona central (A) desplazada hacia la derecha en (a) y la izquierda en (b). El resto se ha llenado con más textura (X e Y). Construyendo tal estereograma se observa un cuadrado central flotando por encima del fondo como se muestra en (c).	307
A.15.El acoplamiento que se ve es el mostrado por los puntos negros	308
A.16.Campo de flujo óptico para un piloto aterrizando un avión	310
A.17.La demostración de los puntos iluminados	311
A.18.(a) Una imagen consistente en un cuadro que se mueve contra un fondo estacionario. (b) Su representación en el campo de velocidad, especificando la velocidad y dirección del movimiento en cada región de la imagen	312
A.19.Gradientes de intensidad de luz pasando sobre una fila de receptores	313
A.20.Cruces por cero y sus derivadas con respecto al tiempo cuando se mueven hacia la derecha (b) y la izquierda (c)	315
A.21.Esbozo $2\frac{1}{2}D$ de un cubo. Las superficies del cubo están representadas por una serie de vectores primitivos como agujas. La longitud representa el grado de inclinación de la superficie, y la orientación de la aguja, la dirección en la que se inclina.	317
A.22.diferentes formas de la letra A	318
A.23.Ejemplo de cono generalizado	320
A.24.Jerarquía de modelos 3D. cada encuadre muestra, a la izquierda el eje principal de la figura y a la derecha sus ejes componentes	321
A.25.Catálogo de descripciones de modelo 3D con diferentes niveles de especificidad	323
B.1. De escena 3-D a la imagen 2-D	326
B.2. intensidad	328

B.3. Intensidad-Illuminancia	329
B.4. Fuentes no puntuales	330
B.5. Emisores y Difusores	331
B.6. Relaciones fotométricas entre superficies	332
B.7. Relación flujo-lumimancia	334

Índice de cuadros

9.1. R-Tabla	208
12.1. Posibles parámetros para el predictor.	274
12.2. Ratios en bits por pixel para diferentes algoritmos de codificación de imágenes en color con calidad de difusión	280
12.3. Tabla de cuantificación de la luminancia propuesta por ISO	284

Capítulo 1

Introducción

Si queremos que nuestro objetivo de crear un autómata antropomorfo autónomo tenga éxito es obvio que hemos de ser capaces de dotarlo de capacidades de percepción. Han pasado más de dos décadas desde que varios investigadores y grupos de investigación comenzaron a intentar automatizar el proceso de visión humana, y sin embargo prevalece un cierto sentido de frustración. Es obvio que esta frustración existe debido, entre otras cosas, a la propia sofisticación del sistema de percepción humana. Sin embargo, hemos de tener en cuenta que estas habilidades son el resultado de millones de años de evolución y que nuestros sistemas de percepción no son infalibles, como muestra la clásica espiral de Fraser (ver figura 1.2). No obstante, la impaciencia está justificada. Tal vez, la investigación en visión artificial ha sido altamente improductiva o peor aún, el esquema de trabajo adoptado por la comunidad de investigadores es erróneo.

1.1. ¿Qué es la visión artificial?

Tal y como se define en [38], podríamos decir que *la visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo*. Las imágenes pueden ser monocromáticas (de niveles de gris) o multibanda, pueden provenir de una o varias cámaras e incluso cada cámara puede ser fija o móvil.

Las estructuras y propiedades del mundo tridimensional que queremos deducir en visión artificial incluyen no sólo sus propiedades geométricas, sino también sus propiedades

materiales. Ejemplos de propiedades geométricas son la forma, tamaño y localización de los objetos. Ejemplos de propiedades de los materiales son su color, iluminación, textura y composición. Si el mundo se modifica en el proceso de formación de la imagen, necesitaremos inferir también la naturaleza del cambio, e incluso predecir el futuro.

¿Por qué es tan difícil el problema de la visión artificial?. Una de las razones es que la formación de imágenes es una aplicación de muchas y diferentes escenas tridimensionales en una imagen bidimensional. Como resultado, el problema inverso, dada una imagen bidimensional determinar la escena tridimensional de la que procede, no tiene solución única. Equivalentemente, de una única imagen de una escena no es posible recobrar sin ambigüedades la escena de la que proviene. Hay varias posibles soluciones a este problema: (1) tomar más imágenes y (2) hacer más hipótesis sobre el mundo. En este segundo caso es importante que cualquier hipótesis que hagamos sea mantenible y que entendamos exactamente su papel.

Incluso cuando el problema de inversión de la imagen esté suficientemente restringido de forma que casi tengamos una solución única nos quedan dos problemas, su computabilidad y su robustez. ¿Es la solución computable usando capacidades de procesamiento razonables?, y además ¿es dicha solución robusta?, en el sentido de admitir pequeños errores bien en los datos o en la propia computación.

Nos podemos preguntar si la visión artificial no es una forma de procesamiento de imágenes o reconocimiento de patrones. Aunque existe algún solapamiento, las diferencias son muy importantes. Procesamiento de imágenes es un término genérico para la manipulación de imágenes con objeto de producir nuevas imágenes que son mejores, en algún sentido. El procesamiento de imágenes comprende: la mejora de imágenes, que modifica una imagen para mejorar su apariencia a nivel humano; la restauración de imágenes que corrige la degradación de imágenes, por ejemplo por emborronamiento por movimiento; y compresión de imágenes, que pretende la representación de imágenes de forma compacta manteniendo un cierto nivel de calidad. Por su parte, el reconocimiento de patrones, o clasificación de patrones, clasifica patrones en un elemento de un conjunto finito de posibles categorías. En general, podría decirse, para simplificar, que el énfasis en el reconocimiento de patrones está sobre patrones bidimensionales como letras del alfabeto. La visión artificial, por el contrario, se dedica más a la generación de descripciones de escenas tridimensionales -que no están, en principio, restringidas a pertenecer a unas clases predeterminadas- a partir de imágenes bidimensionales.

El propósito de la visión artificial es inferir el estado del mundo físico a partir de

imágenes ruidosas y ambiguas del mundo. De las muchas fuentes de información en una escena, sólo unas pocas están presentes en la imagen. Además de la información proporcionada por las propias imágenes, a veces es posible tener conocimiento sobre los objetos, su conducta y el contexto. Otro aspecto importante es el que se refiere a la captación pasiva de la imagen, es decir no influir sobre las condiciones de luminosidad, en lugar de trabajar en lo que podríamos llamar captación o sensado activo.

La aproximación a la visión artificial que cuenta con un mayor número de adeptos es la que recibe el nombre de modular, como alternativa a la que por ejemplo tiene como objetivo realizar una optimización global. En la aproximación modular el objetivo es primero aislar las diferentes fuentes de información en una imagen: estas fuentes pueden ser por ejemplo discontinuidades, sombreado y movimiento. Entonces se pueden usar bien en solitario o en combinación estas fuentes para hacer inferencias sobre la escena. Es obvio, como bien sabemos, que un sistema modular es más fácil de controlar, monitorizar y mejorar así como de comprender.

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

La figura 1.1 muestra un esquema simple de un sistema de visión de propósito general. Los recuadros en la figura representan datos y las flechas el flujo de los datos.

En este modelo, la escena tridimensional es observada primero por una o varias cámaras para producir una o varias imágenes monocromáticas o multibanda. Se detectan entonces las discontinuidades en la luminosidad y estos elementos reciben el nombre de bordes. Puede ocurrir que la segmentación se produzca también en regiones relativamente homogéneas. Estas dos informaciones se usan para restringir las posibles interpretaciones de la escena. Para restringir la información sobre la escena se usan varias características como: mapas de bordes, sombreado, textura, estereo y movimiento. Las restricciones a las que dan lugar estas características pueden ser bien locales (orientación, reflectancia de puntos) o globales (por ejemplo el objeto es de tal tipo.). Todas las restricciones simultáneamente generan descripciones de la imagen que particionan la imagen que pueden servir con varios objetivos (ver [28]).

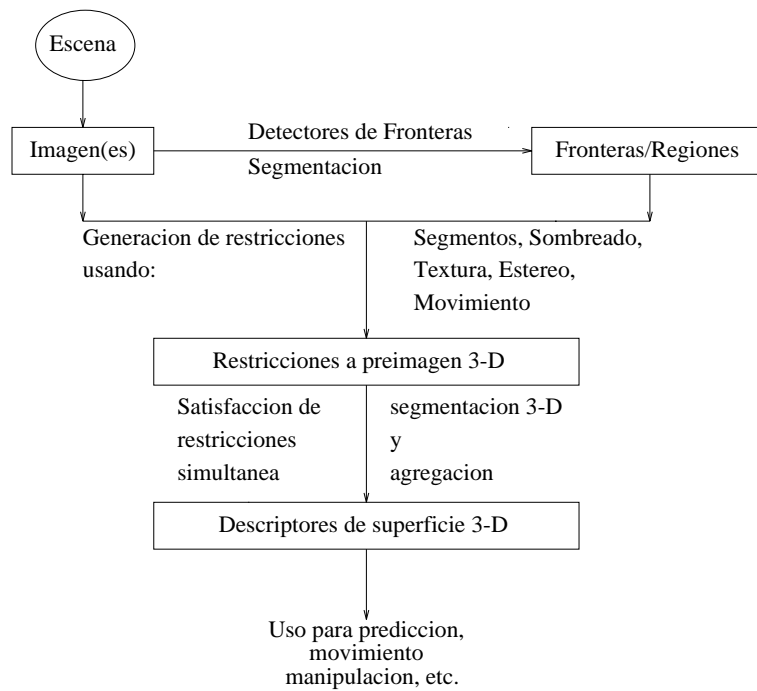


Figura 1.1: Sistema de visión de propósito general

1.2. Nuestro Patrocinador: El Ojo

Una de las preguntas que uno puede hacerse es por qué los investigadores en visión artificial no construyen simplemente un sistema que emule el sistema visual humano, teniendo en cuenta la enorme cantidad de publicaciones en neurofisiología, psicología y psicofísica. Una buena razón por la que los investigadores en visión artificial no emulan la visión humana es porque lo que se conoce del sistema de visión humano más allá del propio ojo es principalmente disjunto, especulativo y escaso. Pero hay más, aunque el sistema de visión humano es adecuado para muchas tareas, es obvio, como veremos que adecuado no equivale a infalible. La no infalibilidad del sistema es ampliamente demostrada por la existencia de ilusiones visuales. Una de ellas se ilustra en la figura 1.2.

Antes de seguir adelante, describiendo el ojo humano, la retina y los caminos visuales al cerebro, discutamos un momento sobre la percepción y el conocimiento. Aunque es obvio que nadie puede negar el papel fundamental que juega la percepción en la adquisición de información por los humanos, podría argumentarse que ver es un acto mecánico y que no genera nada. Todo lo que el acto de ver hace es inferir el estado del mundo en la medida de lo que es permitido por los datos sensados. Es decir, produce alimento para el pensamiento,

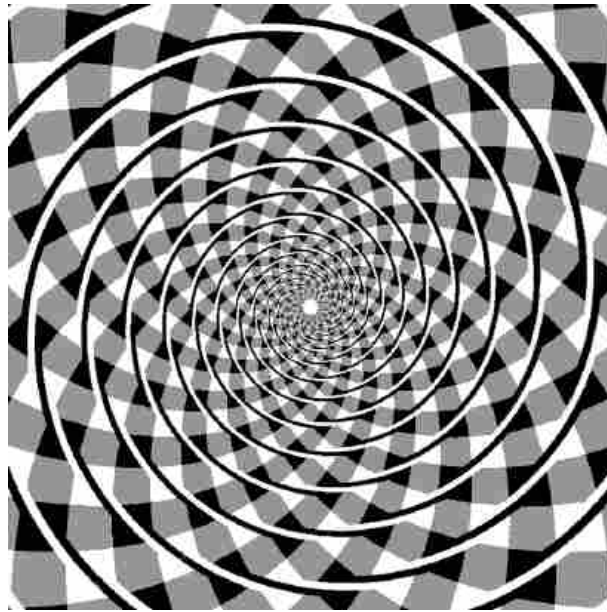


Figura 1.2: La espiral de Fraser. Ilusión de espirales

para conceptualizar y clasificar -asignar pertenencia a una clase de equivalencia basada en formas o funciones, por tanto adscribiendo propiedades que no son percibidas y tan solo postuladas. Dejemos este tema pero pensemos sobre él.

Con independencia de si queremos emular o no la visión humana -bien en la forma o en el funcionamiento- sería bueno preguntarnos qué conocemos del sistema de visión humano.

El ojo humano es aproximadamente un globo de 2 cms de diámetro que puede rotar en su órbita bajo el control de seis músculos extrínsecos. La figura 1.3 muestra una sección del ojo derecho humano. La luz entra a través de la transparente córnea, pasa a través del humor acuoso que rellena la cámara anterior, llega al cristalino y entonces a través del gelatinoso humor vítreo forma una imagen invertida en la retina. Delante del cristalino tenemos una membrana opaca llamada iris que da al ojo su color. La luz sólo puede entrar al ojo a través de una apertura circular del iris que se llama pupila, cuyo tamaño se controla por la expansión y contracción del iris. El cristalino se mantiene en su sitio por los ligamentos suspensorios a través de los cuales el músculo ciliar ajusta la curvatura del cristalino.

El ajuste de la curvatura del cristalino de un ojo por su músculo ciliar se llama acomodación. La acomodación adapta el ojo para obtener la mejor visión a una distancia

concreta, (en los ojos sanos, los objetos a esa distancia se forman en la retina, ni delante ni detrás). Cuando los objetos se forman delante o detrás de la retina aparecen borrosos al observador. La miopía describe la incapacidad del ojo para enfocar en la retina objetos distantes del ojo, en este caso los objetos se forman delante de la retina. La hipermetropía describe la incapacidad de un ojo para enfocar en la retina objetos que están próximos al ojo, (estos objetos se forman detrás de la retina).

1.2.1. La retina

La retina es una membrana nerviosa compleja con un mosaico de fotorreceptores que cuando es estimulada por la luz produce señales eléctricas nerviosas. Estas señales se transmiten al cerebro a través del nervio óptico, evocando la experiencia de la visión. El lugar de la retina donde se juntan todas las fibras nerviosas individuales que constituyen el nervio óptico recibe el nombre de disco óptico. Esta región no tiene fotorreceptores y por eso se llama a menudo punto ciego. No lejos del punto ciego existe una zona con una alta concentración de fotorreceptores llamada la fovea, con un diámetro menor que un milímetro, la fovea tiene una importancia enorme ya que proporciona la agudeza visual en escenas fuertemente iluminadas. Los objetos se forman en la fovea cuando dirigimos nuestra mirada hacia ellos.

La retina, que es menos de un milímetro de gruesa, ha sido el objeto de numerosos estudios. El primer punto importante a notar es que las fibras del nervio óptico se colocan cerca del humor vítreo y sin embargo, los bastones y conos -los dos tipos de receptores, que reciben su nombre de su forma- están localizados cerca de la coroides. Esta disposición es obviamente contraintuitiva ya que obliga a que la luz pase a través de casi toda la profundidad del tejido de la retina -lleno de capilares sanguíneos, células nerviosas y fibras nerviosas- antes de que pueda ser sensada. Sin embargo en la fovea, al contrario que en el resto de la retina, mucho del tejido retinal entre los fotorreceptores y el humor vítreo está desplazado hacia un lado creando una zona con acceso directo a la luz.

La figura 1.4 muestra cómo la luz llega a la retina a través del humor vítreo que está en la parte inferior de la imagen y luego pasa a través del tejido retinal para ser detectado por los fotorreceptores que son los bastones y los conos. Los bastones, que son capaces de detectar luz mucho más débil, facilitan la visión en la oscuridad. Por otra parte, los conos, que al contrario de los bastones tienen diferentes variedades, cada variedad con una sensibilidad preferentes hacia la luz verde, roja o azul nos proporcionan la visión en color.

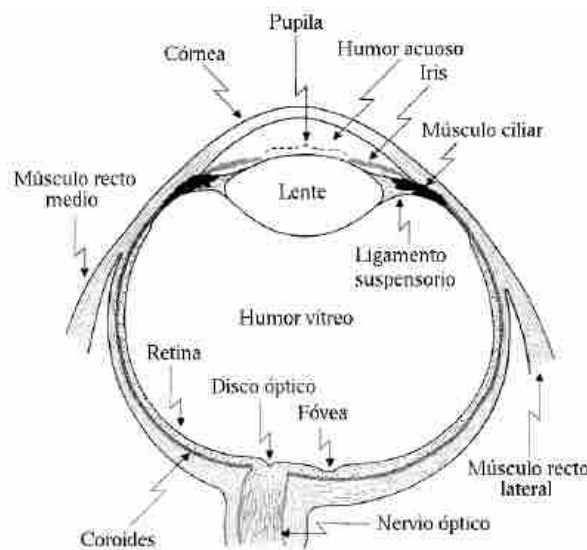


Figura 1.3: El ojo humano

La figura 1.5 muestra la distribución de conos y bastones en la retina humana. En la capa sináptica más exterior, los conos y bastones terminales forman sinapsis con las células bipolares. Una sinapsis es un lugar en el cual una célula del sistema nervioso transmite señal a otra. Las células bipolares llevan a las células ganglionares con las que forman sinapsis en la capa sináptica interna. Es la fibra nerviosa de las células ganglionares la que llega a formar el nervio óptico que lleva al cerebro. El número de fibras ópticas es alrededor de un millón mientras que el número aproximado de conos y bastones es de 125 millones. Es obvio que la señal generada en los conos y bastones es codificada y compactada antes de llegar al cerebro. Esta compactación se lleva a cabo con la ayuda de células amacrinas y horizontales que proporcionan conexiones sinápticas laterales en las capas interna y externa, respectivamente.

1.2.2. Los caminos visuales al cerebro

Aunque en los dos siglos pasados se ha visto sustancial mejora en nuestra comprensión de la estructura y funcionamiento del ojo humano, las partes más centrales del sistema visual tienen muy poco que no sea un misterio de momento. Es la conversión de las imágenes retinianas en conocimiento del mundo lo que forma la auténtica barrera para nuestro entendimiento de la visión humana. Aunque tenemos idea de los caminos visuales del ojo al cerebro, sabemos poco de lo que ocurre en el cerebro.

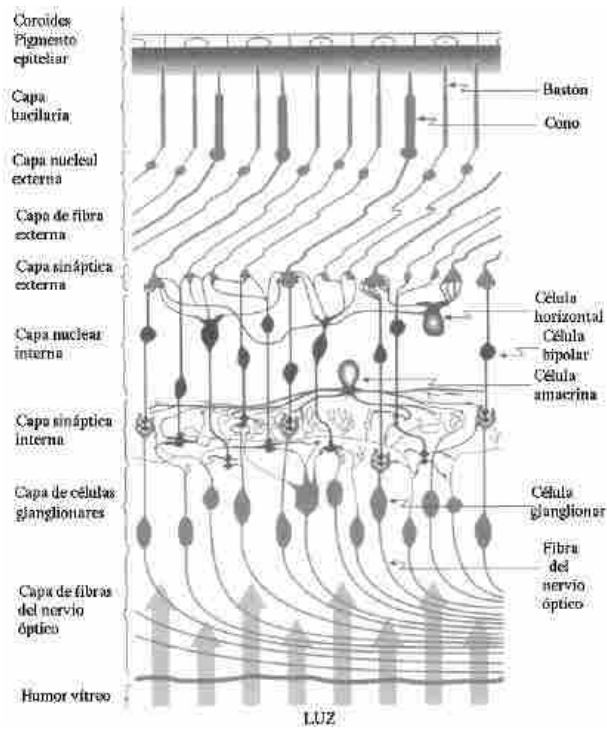


Figura 1.4: La retina

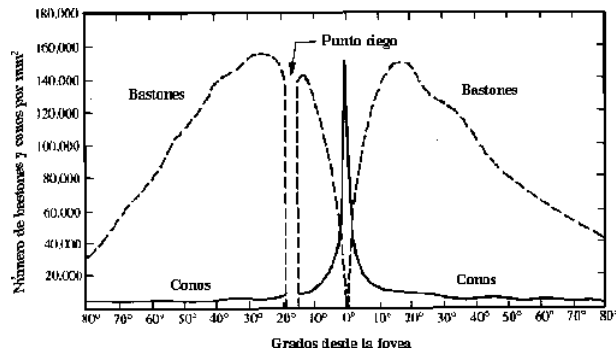


Figura 1.5: Distribución de conos y bastones

La figura 1.6 ilustra los mayores caminos visuales del ojo al cerebro. De cada ojo emerge un nervio óptico que transporta señales eléctricas nerviosas del ojo al cerebro. Las fibras que constituyen cada nervio óptico pueden dividirse en dos grupos: las más cercanas a las fosas nasales y las más exteriores. Las más exteriores van al mismo lugar del cerebro que en el que se originaron mientras que las más próximas a la nariz cruzan en el quiasma óptico para ir al lado contrario del cerebro al que estaban. Así pues como vemos en la figura 1.6 el campo visual izquierdo va al lado derecho del cerebro y el derecho en el izquierdo. Lo que es importante es que las dos imágenes retinianas de cualquier punto de la escena que es visible a ambos ojos se localizan en la misma región del cerebro. Es la disparidad entre las dos imágenes retinianas la que hace posible la percepción de la profundidad estereoscópica dentro del campo de visión que comparten los dos ojos (El término estereoscópico significa ver los objetos en tres dimensiones debido a la diferencia entre las imágenes formadas en los dos ojos; es importante que tengamos en cuenta que es posible percibir un objeto en tres dimensiones sin el uso de ambos ojos).

Para entender el significado de la visión estereoscópica basta con intentar colocarle la caperuza, con una mano, a un bolígrafo que sujetamos con la otra mano a una cierta distancia delante nuestra. Veremos que es mucho más difícil cuando lo intentamos con un ojo cerrado.

Volviendo a los caminos visuales, a partir del quiasma óptico las fibras ópticas siguen en dos grupos llamados los tractos ópticos. Cada tracto comprende fibras que proceden de las mismas partes de las dos retinas. La mayoría de las fibras proceden hasta los núcleos geniculados externos que es una región donde se producen las últimas sinapsis. De los dos núcleos geniculados externos las señales son transmitidas vía las radiaciones ópticas a la corteza estriada que recibe este nombre por su apariencia a rayas en el cerebro humano. La importancia de la fovea se manifiesta en la corteza estriada por el tamaño del área que le dedica en ella. No se sabe mucho de lo que pasa en las regiones de la mente dedicadas a la visión.

1.3. Ilusiones, Ambigüedades e Inconsistencias

Como ya hemos visto la falibilidad del sistema de visión humano ha sido demostrada ampliamente por la existencia de ilusiones visuales, ambigüedades e inconsistencias ([10], [14]).

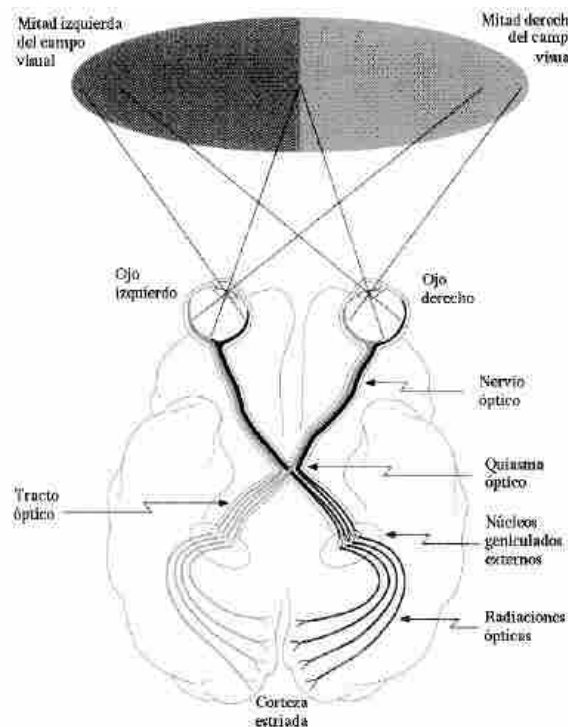


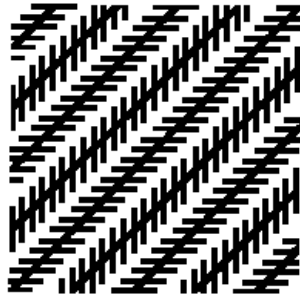
Figura 1.6: Caminos visuales del ojo al cerebro

Uno de los ejemplos de ilusión es la espiral de Fraser que ya hemos visto. Existen otros ejemplos, como los que se muestran en la figura 1.7 de ilusiones ópticas.

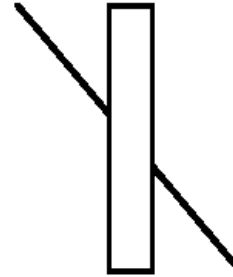
Otro problema es que una figura puede tener más de una posible interpretación, en otras palabras, la figura puede ser ambigua. Múltiples interpretaciones pueden coexistir o una puede dominar a la(s) otra(s). Dos ejemplos son las figuras 1.8,1.9

Finalmente, es posible que, aunque percibida como una figura que ni es ambigua ni corresponde a una ilusión óptica, la figura sea completamente irrealizable en el sentido de que no podemos construir el objeto $3 - D$ percibido en el espacio real $3 - D$. Esta posibilidad es ilustrada bellamente por Belvedere, la famosa litografía de Escher que reproducimos en la figura 1.10. Aunque la litografía se percibe al principio como una escena normal, un examen más detallado de la misma revela que la escena percibida es físicamente irrealizable.

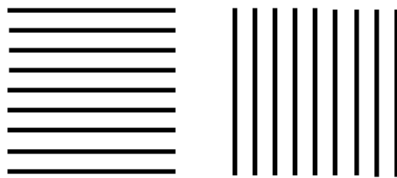
Las diferentes ilusiones, ambigüedades e inconsistencias que hemos analizado son mucho más que curiosidades. Nos llevan a preguntarnos si inferimos de nuestras imágenes en la retina más de lo que de verdad está soportado por la geometría y física de la formación de imágenes. Helmholtz en su libro *Handbook of Physiological Optics* publicado



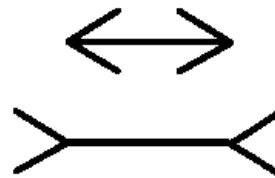
Ilusión de Zöllner (1860)



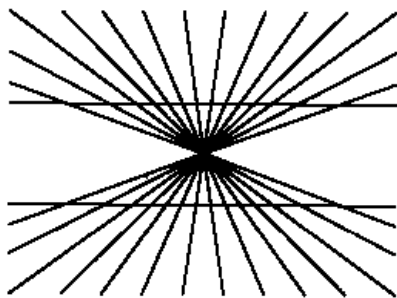
Ilusión de Poggendorff (1860)



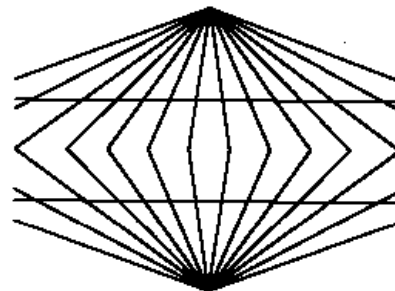
Cuadrados de Helmholtz (1866)



Ilusión de Müller-Lyer (1889)



Ilusión de Hering (1861)



Ilusión de Wundt (1896)

Figura 1.7: Seis ilusiones ópticas clásicas. En la ilusión de Zöllner las diagonales son paralelas pero no lo parecen, en la de Poggendorff, la línea recta no lo parece, en la de Helmholtz los cuadrados parecen rectangulares, en la de Müller-Lyer las líneas horizontales parecen de diferente longitud y en los de Hering y Wundt las líneas rectas parecen curvadas.



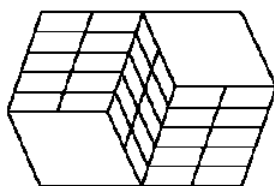
Figura 1.8: Este dibujo indio es ambiguo, tenemos a Poseidón y una sirena



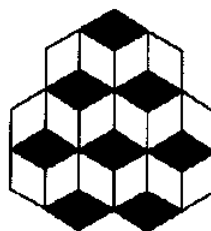
Mujer joven / Anciana



Jarrón / Caras



Hacia dentro / Hacia fuera



Seis cubos / Siete cubos

Figura 1.9: Cuatro ejemplos clásicos de ambigüedad

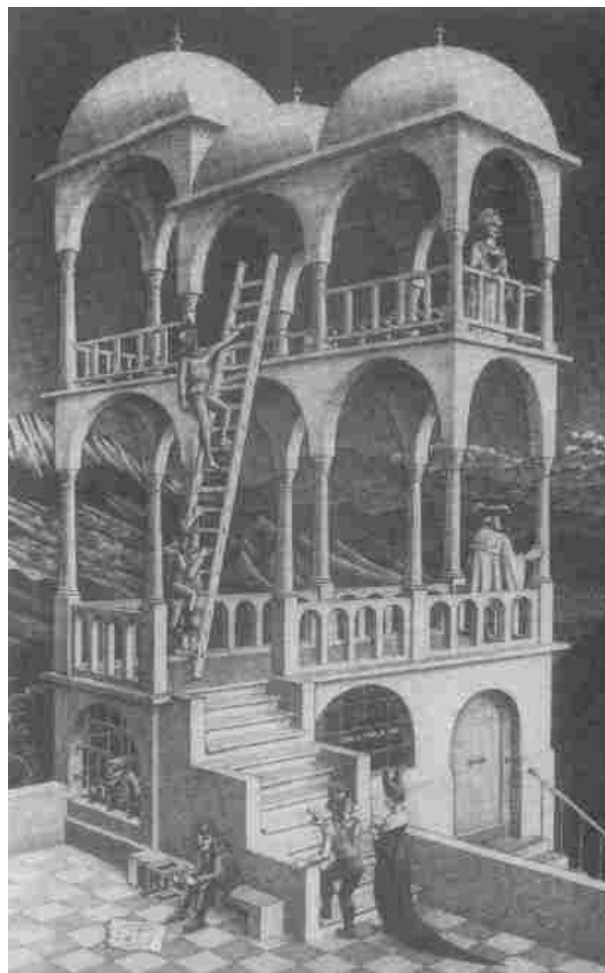


Figura 1.10: Consistencias Geométricas

a mediados del siglo pasado expresó que cada imagen es una imagen de algo sólo para aquel que conoce como leerla, y que está capacitado con la ayuda de la imagen a formar una idea de la cosa. La implicación de esta afirmación es que no es como si el sistema visual humano estuviera haciendo inferencias precisas y exactas basadas en la física de la formación de imágenes en el ojo, sino que el sistema visual invoca reglas que se obtienen y están sesgadas por la experiencia previa del individuo y tal vez por la especie. Como resultado los humanos podrían ver lo que no hay y no ver lo que hay. Una pregunta nos asalta a todos, ¿le permitiríamos a una máquina hacer lo mismo?.

Capítulo 2

Fundamentos

En este tema vamos a analizar algunos conceptos que son fundamentales en el procesamiento de imágenes. La referencia básica será [12].

Comenzaremos analizando brevemente los tipos de imágenes. Veremos a continuación, de una forma simple e intuitiva, la modelización de una imagen y como se obtiene ésta a partir de una escena tridimensional. Posteriormente examinaremos los efectos del muestreo y la cuantificación, para estudiar a continuación las distorsiones que se producen en la imagen. Tras analizar algunas relaciones entre píxeles terminaremos el tema describiendo un sistema de procesamiento de imágenes de propósito general.

2.1. Elementos de un sistema de procesamiento de imágenes

Los elementos usuales de un sistema de procesamiento de imágenes de tipo general son los siguientes: **Captación**, **Almacenamiento**, **Procesamiento**, **Comunicaciones** y **Visualización**.

Ninguno de estos elementos será descrito en este tema, por lo que se recomienda la visita a algún centro de instrumentación científica (como el de nuestra Universidad) para examinar los componentes de un sistema de procesamiento de imágenes.

Sí parece importante examinar la tabla siguiente que muestra diferentes tipos de imágenes y sensores para la captación de escenas. *bpp* denota bits por pixel, Scan se usa para notar escáner, n.g. nivel de gris, Micr. para microscopio, Micria. para microscopía y electro. para electrónica.

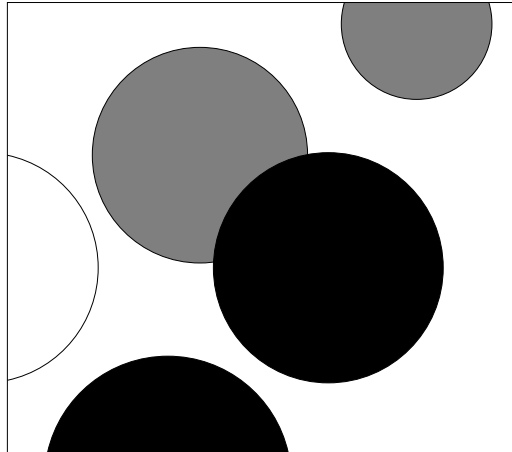


Figura 2.1: Un Mondrian formado por regiones de color constante

<i>Escena</i>		<i>Sensor</i>	<i>Tipo</i>	<i>Bandas</i>	<i>bpp</i>	<i>tamaño</i>
(fuera)dentro		V.C.	n.g./color	1-3(2D)	8-24	768x512
Micr. óptico		Micria. óptica	n.g./color	1-3(2D)	8-24	768x512
		Micria. electro.	n.g.	1(2D)	8	(Film)
Tierra		Scan.+Satélite.	n.g.	>4(2D)	8	>3000x2000
Cielo		Telesc+CCD	Intensit.	4(2D)	32	>1400x700
Cuerpo	X-rays	Film	n.g.	1(2D)	8	(Film)
	C.T.	Scan. Barrido	n.g.	1(3D)	8	512 ² x24xseg
	M.N.R.	Scan. Barrido	n.g.	1(3D)	8	256 ² x40
	Eco.	Video	n.g.	1(2D)	6	512 ² x?
Foto		Scan. sobremesa	n.g./color	1-3(2D)	8-24	600 ppp

2.2. Modelo de Imagen Simple

Consideremos una situación simple en la que la orientación de una superficie no juega ningún papel, por ejemplo un grupo de círculos en el que dentro de cada uno se refleja la luz de la misma forma (ver figura 2.1). Este tipo de escena recibe el nombre de Mondrian en honor del pintor holandés Cornelis Mondrain que a comienzos del siglo veinte pintaba cuadros con estas características.

En una situación de este tipo, la radiancia de la escena, que se nota $l(x, y)$, se carac-

teriza por dos componentes: (1) la cantidad de luz que incide en la escena que se está observando y (2) la cantidad de luz que reflejan los objetos en la escena. Estos conceptos reciben el nombre de *irradiación* y *reflectancia* y se notan $e(x, y)$ y $r(x, y)$ respectivamente. Su producto define $l(x, y)$, es decir

$$l(x, y) = e(x, y)r(x, y)$$

donde

$$0 < e(x, y) < \infty \quad \text{y} \quad 0 < r(x, y) < 1$$

para clarificar conceptos observemos que e mide energía que le llega, mientras que l mide la energía que emite. Normalmente, la energía que le llega proviene de un punto de luz.

Las ecuaciones anteriores indican que la reflectancia está acotada por 0 (absorción total) y 1 (reflectancia total). La naturaleza de $e(x, y)$ se determina por la fuente de luz y $r(x, y)$ lo determina la característica del objeto.

Algunos ejemplos típicos son los siguientes. En un día claro la iluminación puede ser de 9000 candelas en la superficie de la Tierra y menor que 1000 en un día nublado. La luna puede producir 0.01 candelas de iluminación. En cuanto a la reflectancia 0,01 es el número correspondiente a terciopelo negro, 0,65 le corresponde a acero inoxidable, 0,80 para las paredes blancas, 0,93 para la nieve, etc.

Es evidente de las ecuaciones anteriores que

$$L_{\min} \leq l \leq L_{\max}$$

Para interiores se puede tener que $L_{\min} \approx 0,005$ y $L_{\max} \approx 100$. $[L_{\min}, L_{\max}]$ recibe el nombre de escala de niveles de gris. Usualmente, este rango se lleva, en el procesamiento posterior, al intervalo $[0, L]$ donde $l = 0$ se considera negro y $l = L$ se considera blanco.

Es obvio que una pregunta a la que necesitamos dar respuesta es qué relación existe entre la radiancia en un punto de un objeto (radiación de la escena) L y la irradiación en el correspondiente punto de la imagen captada (irradiación de la imagen) E . Si d es el diámetro de la lente y f es la distancia al plano de la imagen, puede probarse que (ver figura B.1)

$$E = L \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha$$

donde α es el ángulo del rayo desde el objeto hasta el centro de la lente. (ver [18]). Es importante no confundir esta f con una imagen, aquí se usa como distancia focal.

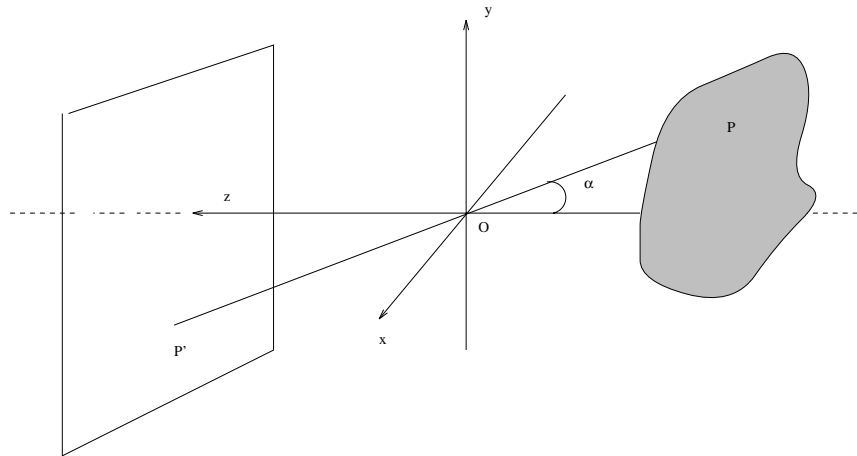


Figura 2.2: De escena 3-D a la imagen 2-D

2.3. Geometría de Imágenes

Estudiaremos ahora como se forman las imágenes 2-D a partir de las escenas tridimensionales. Entender la formación de imágenes es un prerequisite para entender completamente los métodos de recuperar información 3-D a partir de imágenes. Si entendemos el proceso por el cual el mundo 3-D (una escena) se proyecta en el mundo 2-D (una imagen) comprenderemos dos aspectos fundamentales de la formación de imágenes:

- ¿Qué determina donde aparece la imagen de un punto?.
- ¿Qué determina lo brillante que la imagen de una superficie es?.

La respuesta a la segunda pregunta ha sido proporcionada de una forma simple en la sección anterior. Veamos ahora la respuesta a la primera.

2.3.1. Perspectiva por Proyección

Consideremos un agujero diminuto ideal (ojo de aguja) a una distancia fija delante del plano de la imagen (ver figura B.1). Supongamos que idealmente sólo la luz que viene a partir del ojo de la aguja puede alcanzar el plano de la imagen. Puesto que la luz viaja en línea recta, cada punto en la imagen corresponde a una dirección particular definida por un rayo desde el punto a través del ojo de la aguja.

Definimos el eje óptico, en este caso simple, como la perpendicular, a través del ojo de la aguja, al plano de la imagen. Introducimos un sistema de coordenadas con el origen en el ojo de la aguja y el eje z alineado con el eje óptico y apuntando hacia la imagen. Es importante tener en cuenta que con esta orientación los puntos hacia los que mira la cámara tienen componente z negativa.

Queremos ahora calcular dónde aparecerá la imagen P' del punto P que está sobre un objeto enfrente de la cámara (ver figura B.1). Supondremos que no existe ningún otro objeto en el rayo que une P y el ojo de la aguja O . Sea $\mathbf{r} = (x, y, z)^t$ el vector que une P y O y $\mathbf{r}' = (x', y', f')$ el que une O y P' .

Si el rayo que conecta P y P' forma un ángulo α con el eje óptico tendremos

$$\cos(\alpha) = -\frac{\mathbf{r}\hat{\mathbf{z}}}{\|\mathbf{r}\|}$$

donde $\hat{\mathbf{z}}$ es el vector unitario en la dirección del eje óptico. Es importante recordar que los puntos delante de la cámara tienen componente z negativa. Además

$$\cos(\alpha) = \frac{f'}{\|\mathbf{r}'\|}$$

de modo que

$$\frac{\|\mathbf{r}'\|}{f'} = -\frac{\|\mathbf{r}\|}{\mathbf{r}\hat{\mathbf{z}}} \quad (2.1)$$

lo que en componentes significa

$$\frac{x'}{f'} = \frac{x}{z} \quad \text{y} \quad \frac{y'}{f'} = \frac{y}{z}.$$

El mismo modelo puede obtenerse cuando el plano $z = 0$ se situa en el plano de la imagen como se detalla en [12].

2.3.2. Proyección Ortográfica

Consideremos la proyección de perspectiva donde el plano de la imagen se encuentra en $z = 0$, el ojo de la aguja en $z = f$ y el plano de la escena en $z = z$, con $f > z$. En ese caso las ecuaciones de la proyección de perspectiva serían, observemos que en este caso las imágenes no salen invertidas,

$$\frac{x'}{f} = \frac{x}{f - z} \quad \text{y} \quad \frac{y'}{f} = \frac{y}{f - z}.$$

obviamente la z proyectada tiene $z = 0$ en todos los casos. Observemos que cuando f tiende a infinito tenemos que $x = x'$ e $y = y'$. Llamaremos perspectiva con distorsión a la que aplica (x, y, z) en $(fx/(f - z), fy/(f - z), fz/(f - z))$. Esta transformación recibe el nombre de proyección ortográfica cuando el ojo de la aguja está a distancia infinito en la dirección de z .

La perspectiva con distorsión produce un objeto tridimensional al que se le ha quitado la forma, cuanto más lejano del ojo de la aguja esté más pequeño se verá.

Observemos que en los modelos que hemos descrito hemos perdido la información sobre el eje z . Es claro que para recuperarla necesitamos al menos dos imágenes distintas, desde diferentes ángulos, del mismo objeto.

2.4. Efectos del muestreo y la cuantificación

Las dos principales causas que producen pérdida de información cuando capturamos una imagen son la naturaleza discreta de los píxeles de la imagen y el rango limitado de valores de intensidad luminosa que somos capaces de medir en cada píxel.

2.4.1. Efectos del muestreo

El muestreo de una imagen tiene el efecto de reducir la resolución espacial de la misma. La figura 4.9 muestra el efecto de captar una misma imagen a distintas resoluciones. Todas ellas han sido transformadas al mismo tamaño para su visualización.

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

2.4.2. Efecto de la cuantificación

El efecto de cuantificación viene dado por la imposibilidad de tener un rango infinito de valores de medida para la intensidad de brillo de los píxeles. La tecnología actual permite en algunos casos llegar hasta 10 bits de información, aunque lo general es tener 8 bit, o equivalentemente 256 niveles de gris para codificar este valor lumínico. La figura 2.4 muestra el efecto de representar una imagen con distintos números de bits (8 y 1).

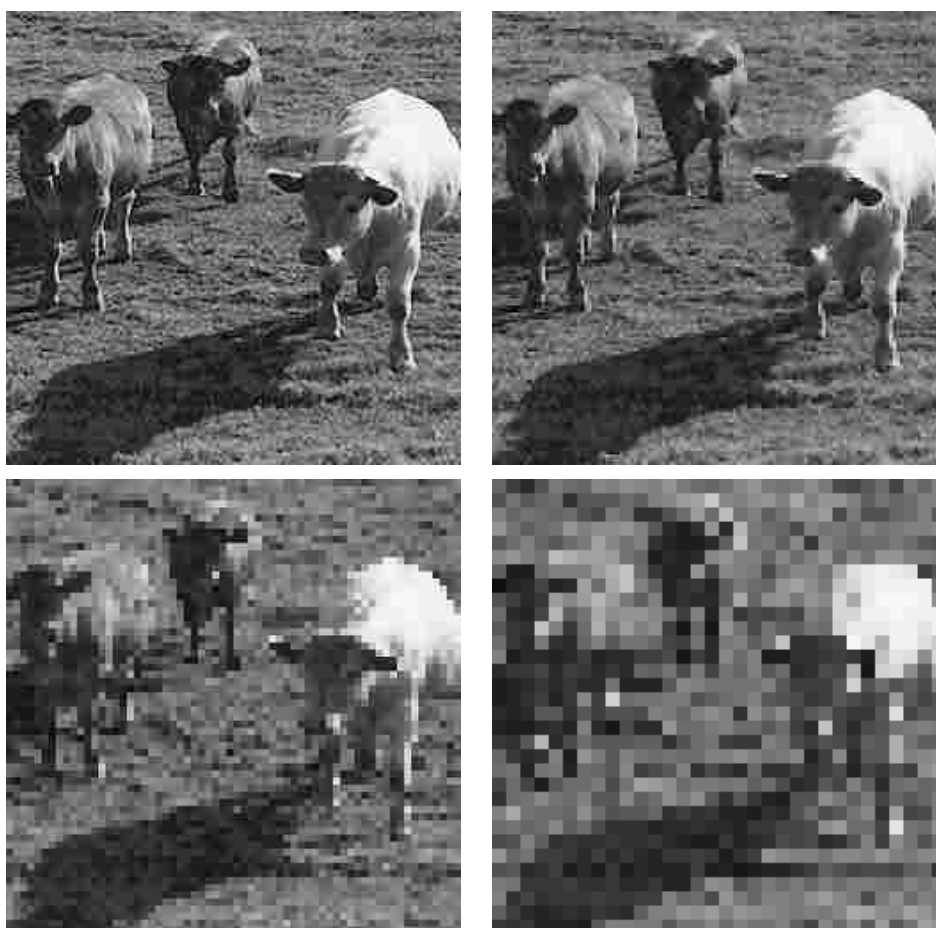


Figura 2.3: De izquierda a derecha y de arriba a abajo. Muestreo a 256×256 , 128×128 , 64×64 y 32×32 píxeles



Figura 2.4: Efecto de la cuantificación. De izquierda a derecha, 8 y 1 bits.

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

2.5. Formación de la Imagen

Supongamos que un objeto tridimensional o escena es convertido en una imagen 2-D mediante un sistema de grabación como una cámara (ver figura 2.5). Los sistemas de grabación no son perfectos e introducen distorsiones estadísticas y determinísticas.

Notaremos por $f(i, j)$ y $g(i, j)$ la imagen original sin distorsiones y la imagen observada, ruidosa y emborronada, respectivamente. Un modelo general de formación de imágenes consta de las partes siguientes

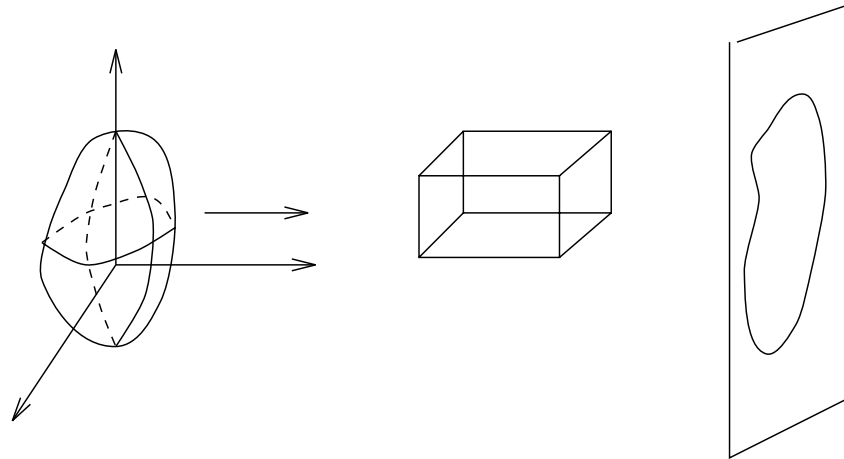


Figura 2.5: Captación de Imagen

Función de Emborronamiento o psf

$$u(i, j) = \sum_m \sum_n d(i, j, m, n) f(m, n) \quad (2.2)$$

Transformación

$$v(i, j) = S(u(i, j)) \quad (2.3)$$

Proceso de Ruido

$$g(i, j) = v(i, j) \odot \epsilon(i, j) \quad (2.4)$$

donde $S\{\cdot\}$ representa una función no lineal, $d(i, j, m, n)$ es la respuesta del sistema de emborronamiento a un impulso bidimensional en la posición (i, j) , $\epsilon(i, j)$ denota el proceso de ruido y \odot representa una operación punto a punto.

Las operaciones descritas son aproximaciones discretas del problema continuo. Un estudio de como se obtiene la versión discreta del problema a partir del continuo, así como las condiciones bajo las cuales la versión continua puede recuperarse a partir de la discreta puede encontrarse en [27], páginas 15 y ss.

Pasaremos ahora a estudiar cada uno de los pasos en la formación de la imagen.

2.5.1. Función de Emborronamiento

El modelo de emborronamiento definido en (2.2) es de uso limitado debido fundamentalmente a las dificultades para estimar $d(i, j, m, n)$ así como al gran esfuerzo computa-

cional que se requiere para las técnicas de restauración a las que da lugar. En la mayoría de las situaciones, el emborronamiento puede ser considerado, al menos localmente, como invariante en el espacio pudiendo entonces escribirse

$$u(i, j) = \sum_m \sum_n d(i - m, j - n) f(m, n) \quad (2.5)$$

Estudiaremos ahora las funciones de emborronamiento más comunes. En este estudio tendremos en cuenta que el proceso de emborronamiento es un proceso continuo y nos propondremos encontrar $d(m, n)$ a partir de su versión continua $d(s, t; \psi)$, donde ψ es un vector que determina la fuerza del emborronamiento. Usaremos la siguiente aproximación:

$$d(m, n) = \int_{\Omega(m, n)} d(s, t; \psi) ds dt \quad (2.6)$$

donde

$$\Omega(m, n) = \left\{ s, t \mid m - \frac{1}{2} \leq s \leq m + \frac{1}{2}, n - \frac{1}{2} \leq t \leq n + \frac{1}{2} \right\}$$

Además habremos de tener en cuenta que la función de emborronamiento no puede tomar valores arbitrarios y que en particular no puede tomar valores negativos y la suma de los coeficientes debe ser igual a uno.

Emborronamiento por Movimiento

Se pueden distinguir muchos tipos de emborronamiento por movimiento, todos ellos son debidos al movimiento relativo entre el aparato que capta la imagen y el objeto. Estos movimientos pueden ser traslaciones, rotaciones, cambios de escala o incluso combinaciones de estos casos.

Si el objeto es trasladado a una velocidad constante V en un ángulo de ϕ radianes con el eje horizontal durante un intervalo de exposición $[0, T]$, la distorsión es unidimensional. Si definimos la *longitud del movimiento* mediante $L = VT$, la psf viene dada por

$$d(s, t; L, \psi) = \begin{cases} 1/L & \text{si } \sqrt{s^2 + t^2} \leq L/2 \text{ y } t/s = \tan(\psi) \\ 0 & \text{en caso contrario} \end{cases}$$

Emborronamiento por desenfoque

Cuando una escena 3-D produce una imagen 2-D algunas partes de la escena están enfocadas, mientras que otras no lo están. Si la apertura de la cámara es circular, la imagen de un punto fuente es un disco pequeño conocido como el círculo de confusión. El



Figura 2.6: Imagen original y borrosa por desenfoco.

grado de desenfoco (el diámetro del círculo de confusión) depende de la longitud focal \mathcal{F} , la apertura y la distancia \mathcal{P} entre cámara y el objeto.

Si el grado de desenfoco es muy grande, es posible usar una función de emborronamiento uniforme dentro del círculo de confusión. La psf de este emborronamiento con un radio R igual al círculo de confusión dividido por dos viene dado por:

$$d(s, t; R) = \begin{cases} \frac{1}{\pi R^2} & \text{si } \sqrt{s^2 + t^2} \leq R \\ 0 & \text{en caso contrario} \end{cases}$$

La figura 2.6 muestra un ejemplo de emborronamiento por desenfoco.

Turbulencia Atmosférica

Aunque el emborronamiento introducido por turbulencia atmosférica depende de muchos factores como temperatura, velocidad del viento y tiempo de exposición, para exposiciones largas la función de emborronamiento puede ser aproximada correctamente por una función gaussiana:

$$d(s, t; \sigma) = C \exp \left\{ -\frac{s^2 + t^2}{2\sigma^2} \right\}$$

donde σ^2 determina la severidad del emborronamiento y C es una constante.

Emborronamiento de Dispersión

Las imágenes por rayos X muestran los detalles mediante la cantidad de radiación que fue absorbida por el objeto que fue irradiado. Por desgracia, la irradiación se dispersa de su camino de incidencia lo que produce una distribución de radiación alrededor de un punto. Aunque hay muchos factores que influyen en la correspondiente psf, una aproximación razonable es la siguiente

$$d(s, t; \beta) = \frac{C}{(1 + (s^2 + t^2)/R^2)^\beta}$$

Las constantes β y R determinan la severidad del emborronamiento y son una función de la distancia entre el detector y el objeto radiado.

2.5.2. Transformación

La respuesta de los detectores y almacenadores de imágenes es usualmente no lineal. Por ejemplo, la respuesta de las películas fotográficas, imágenes de scanners y visualizadores puede escribirse como

$$v(i, j) = \alpha u(i, j)^\beta$$

donde α y β dependen del aparato utilizado. Veamos un ejemplo:

Consideremos una señal de video, las componentes de color aparecen en el modelo de color RGB, para televisiones por ejemplo, como voltajes cuyos valores instantáneos son proporcionales a la iluminación que cae en el correspondiente punto del tubo. La señal de video original que llega a un monitor CRT no es apropiada ya que los tubos de rayos catódicos son instrumentos no lineales, la función de transferencia de un CRT produce una intensidad, I , que es proporcional a alguna potencia del voltaje V , mediante la expresión

$$I \propto \alpha V^\gamma$$

donde γ se encuentra entre 2 y 3 dependiendo del tipo de tubo. Son factores gamma típicos 2,2 para el sistema NSTC y 2,8 para PAL/SECAM.

Para compensar por el procesamiento no lineal en el monitor, los datos RGB son gamma corregidos antes de la transmisión (en la cámara) en lugar de en el receptor mediante la ecuación $I_{\text{transmite}} = I_{\text{recibe}}^{0,45}$

Nosotros supondremos que las transformaciones no-lineales pueden ser eliminadas del problema. Así que nuestros procesos de formación de imágenes sólo constarán de emborronamiento y modelo de ruido. Sin embargo, esto requiere un poco de justificación. Algunas veces es posible aproximar la respuesta no lineal por una lineal tomando una transformación apropiada (por ejemplo logarítmica) o incluso realizando la operación inversa del sensor antes que ningún otro tipo de procesamiento. Este aproximación tiene el inconveniente de que el proceso es aplicado también al ruido, aunque los resultados, en general, no son malos.

2.5.3. Modelos de Ruido

Un modelo general de ruido es el siguiente: la operación \odot , que define el proceso de ruido, es la suma y $\epsilon(i, j)$ tiene la siguiente forma

$$\epsilon(i, j) = \sqrt{v(i, j)}\mu_1(i, j) + \mu_2(i, j)$$

donde μ_1 y μ_2 son ruidos blancos gaussianos independientes de media cero. En algunos casos la formula anterior es una aproximación de la expresión para obtener $g(i, j)$

$$g(i, j) = \mathcal{P}(v(i, j)) + \mu_2(i, j)$$

donde \mathcal{P} denota una distribución de Poisson.

La figura 2.7 muestra un ejemplo de imagen ruidosa con ruido aditivo gaussiano de desviación típica 20.

2.6. Algunas Relaciones Básicas Entre Píxeles

En esta sección vamos a considerar varias relaciones simples aunque importantes entre los píxeles de una imagen digital. A lo largo de este curso usaremos normalmente $f(x, y)$ o $g(x, y)$ para notar una imagen. Para píxeles particulares usaremos letras minúsculas como p y q . Un subconjunto de píxeles de $f(x, y)$ se nota S .

2.6.1. Vecinos de un píxel

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

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

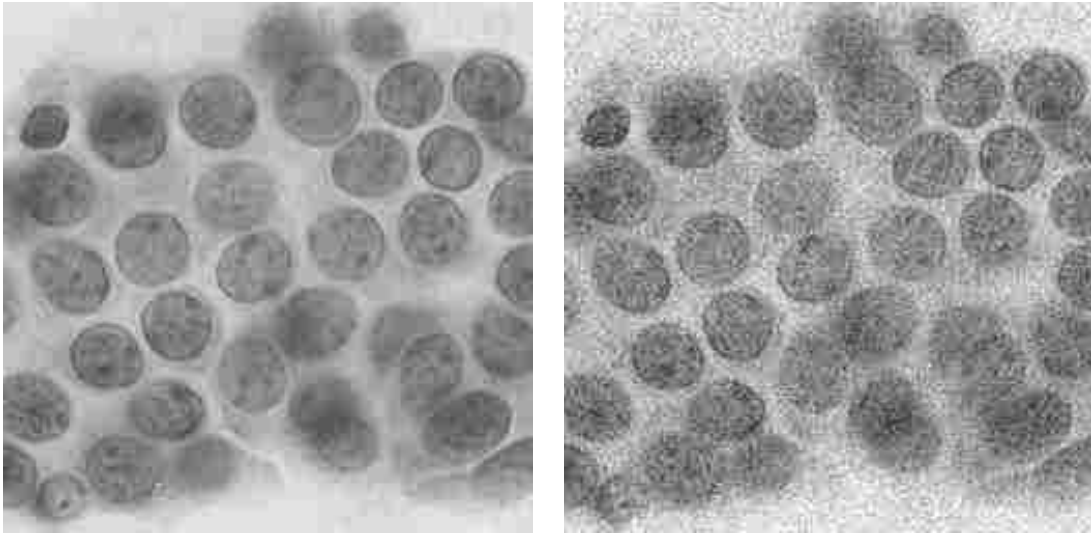


Figura 2.7: Imagen original y ruidosa con ruido gaussiano.

este conjunto de píxeles, que recibe el nombre de 4-vecinos de p , se nota $N_4(p)$. Cada pixel está a distancia unitaria de (x, y) .

Los píxeles diagonales vecinos de p tienen coordenadas

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

y se notan $N_D(p)$. Estos puntos, junto con los cuatro vecinos, se llaman 8 – vecinos de p y se notan $N_8(p)$.

2.6.2. Conectividad

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

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

Consideramos tres tipos de conectividad:

1. 4 – conectividad. Dos píxeles p y q con valores en V se dicen 4-conectados si q pertenece a $N_4(p)$.
2. 8-conectividad. Dos píxeles p y q con valores en V se dicen 8-conectados si q pertenece a $N_8(p)$.
3. m-conectividad. Dos píxeles p y q con valores en V se dicen m-conectados si
 - a) $q \in N_4(p)$ o
 - b) $q \in N_D(p)$ y $N_4(p) \cap N_4(q)$ es vacío. (Este es el conjunto de 4-vecinos de p y q con valores en V).

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

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

Un camino del píxel p con coordenadas (x, y) a q con coordenadas (s, t) es una sucesión distinta de píxeles con coordenadas

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

donde $(x_0, y_0) = (x, y)$ y $(x_n, y_n) = (s, t)$, siendo (x_i, y_i) adyacente a (x_{i-1}, y_{i-1}) para $1 \leq i \leq n$, siendo entonces n la longitud del camino. Es obvio que el tipo de camino depende del tipo de adyacencia utilizado.

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

2.6.3. Medidas de Distancia

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

1. $D(p, q) \geq 0$ ($D(p, q) = 0$ sii $p = q$)

2. $D(p, q) = D(q, p)$ y
3. $D(p, z) \leq D(p, q) + D(q, z)$

La distancia euclídea entre p y q se define como

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

para esta medida de distancia, los píxeles a distancia menor o igual que r de uno dado definen un disco de radio r centrado en él.

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

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

Por ejemplo, los píxeles a distancia menor o igual que 2 de uno dado son los siguientes

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

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

$$D_c(p, q) = \max(|x - s|, |y - t|)$$

por ejemplo los píxeles con distancia D_8 menor o igual que 2 son

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

2.6.4. Operaciones Aritmético / Lógicas

Las operaciones aritmético/lógicas entre píxeles se usan muy frecuentemente en el procesamiento de imágenes. Las operaciones aritméticas entre dos píxeles p y q se notan como sigue

1. Suma: $p + q$,
2. Diferencia: $p - q$,
3. Multiplicación: $p * q$ o pq o $p \times q$
4. División: $p \div q$

Las principales operaciones lógicas entre píxeles son

1. AND: $p\text{AND}q$,
2. OR: $p\text{OR}q$
3. COMPLEMENTO: $\text{NOT}p$

Es importante notar que las operaciones aritméticas se aplican a imágenes multivaluadas, es decir a imágenes cuyos píxeles pueden tomar diferentes valores, mientras que las lógicas se refieren a imágenes bivaluadas tomando los valores 0 o 1.

Además de las operaciones aritmético/lógicas que hemos visto, es posible ampliar estos conceptos, como veremos en capítulos posteriores, a operaciones de máscara en la que se tenga en cuenta los valores de los píxeles vecinos, tanto para imágenes multivaluadas como binarias.

2.7. Pasos Fundamentales en el Procesamiento de Imágenes

La visión artificial lleva asociada una enorme cantidad de conceptos relacionados con hardware, software y también con desarrollos teóricos. En esta sección veremos los pasos fundamentales para llevar a cabo una tarea de visión artificial.

Como ejemplo guía que nos permitirá entender más fácilmente el proceso utilizaremos el problema de reconocimiento de las direcciones en un sobre de correos.

La figura 12.8.2 muestra que el objetivo es producir un resultado para un dominio concreto mediante procesamiento de imágenes, la idea puede extenderse fácilmente a un problema más general. El objetivo es leer la dirección de cada carta. El resultado final será, por tanto, una sucesión de caracteres alfanuméricos.

El primer paso en el proceso es adquirir la imagen digital. Para ello necesitamos sensores y la capacidad para digitalizar la señal producida por el sensor. El sensor puede ser una cámara color o monocromo que produce una imagen completa del dominio del problema cada $1/30$ segundos. Los sensores podrían ser también una cámara scanner que produce una línea en cada instante. En este caso el movimiento del objeto por la línea del scanner (o al revés) produce la imagen bidimensional. Si la salida de la cámara no está en formato digital necesitaríamos usar un conversor analógico digital. La naturaleza del sensor y la imagen que produce vienen determinadas por la aplicación. En el ejemplo que estamos considerando un modelo de captación muy usado es el scanner.

Una vez que la imagen digitalizada ha sido obtenida el siguiente paso consiste en el preprocesamiento de dicha imagen. El objetivo del preprocesamiento es mejorar la imagen de forma que nuestro objetivo final tenga mayores posibilidades de éxito. En nuestro ejemplo, el preprocesamiento trata normalmente con técnicas de mejorar el contraste, suprimir ruido y aislar regiones que por sus características indiquen la posibilidad de información alfanumérica.

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

La salida del proceso de segmentación es una imagen de datos que o bien contienen la frontera de la región o los puntos de ella misma. Necesitamos ahora convertir nuestros datos a una forma que sea apropiada para el ordenador. Nuestra primera decisión es saber si vamos a usar la representación por frontera o región completa. La representación por la frontera es apropiada cuando el objetivo se centra en las características de la forma externa como esquinas o concavidades y convexidades. La representación por regiones es apropiada cuando la atención se centra en propiedades internas como la textura o el esqueleto. Sin embargo, en muchas aplicaciones ambas representaciones coexisten. Esta situación ocurre en nuestro problema donde necesitamos algoritmos basados en la forma de la frontera y también en el esqueleto.

La elección de una representación es sólo una parte de la transformación de los datos de entrada. Necesitamos especificar un método que extraiga los datos de interés. La des-

cripción, que recibe también el nombre de selección de rasgos se dedica a extraer rasgos que producen alguna información cuantitativa de interés o rasgos que son básicos para diferenciar una clase de objetos de otra. En nuestro ejemplo los descriptores pueden ser los agujeros y las concavidades (convexidades) que nos ayudarán a diferenciar una parte del alfabeto de otra.

El último paso de la figura 12.8.2 es el reconocimiento y la interpretación. El reconocimiento es el proceso que asigna una etiqueta a un objeto basada en la información que proporcionan los descriptores. La interpretación lleva a asignar significado al conjunto de objetos reconocidos. En nuestro ejemplo, identificar un caracter como *c* requiere asociar los descriptores para ese caracter con la etiqueta *c*. la interpretación pretende dar sentido a un conjunto de entidades etiquetadas. Por ejemplo, una hilera de cinco números puede ser interpretada como un código.

Hasta ahora no hemos dicho nada sobre la necesidad de conocimiento a priori o sobre la interacción entra la base de conocimiento y los módulos de procesamiento. El conocimiento sobre el dominio del problema se codifica en la forma de una base de conocimiento. Este conocimiento puede ser tan simple como detallar las regiones de la imagen donde la información de interés se encuentra y por tanto ayudando a limitar la búsqueda de los objetos de interés. La base de conocimiento puede ser bastante compleja también, puede incluir una lista de los mayores defectos que se pueden esperar en un problema de inspección de materiales. Además de guiar la operación de cada módulo, la base de conocimientos también controla la interacción entre los diferentes módulos. Esta distinción se hace en la figura 12.8.2 por el uso de flechas de doble sentido que unen la base de conocimiento con los módulos de procesamiento, en contraposición a las flechas unidimensionales entre los módulos de procesamiento. Nuestro formato indica que que la comunicación entre los módulos de procesamiento se basa generalmente en el conocimiento a priori sobre lo que el resultado debería ser. El conocimiento guía no sólo la operación de cada módulo sino también la operación de realimentación entre los módulos a través de la base de conocimiento. Por ejemplo, una hilera de números en la localización correcta pero formada por cuatro caracteres (uno de ellos imposible de reconocer) podría llevar al módulo de interpretación a sospechar que dos caracteres están unidos. Una realimentación a través de la base de conocimiento podría llevar a intentar una nueva segmentación.

Aunque no discutiremos aquí la presentación de imágenes es importante recordar que este proceso puede tener sentido en cualquier paso de la figura 12.8.2. También es importante recordar que no todas las aplicaciones necesitan todas las interacciones que se

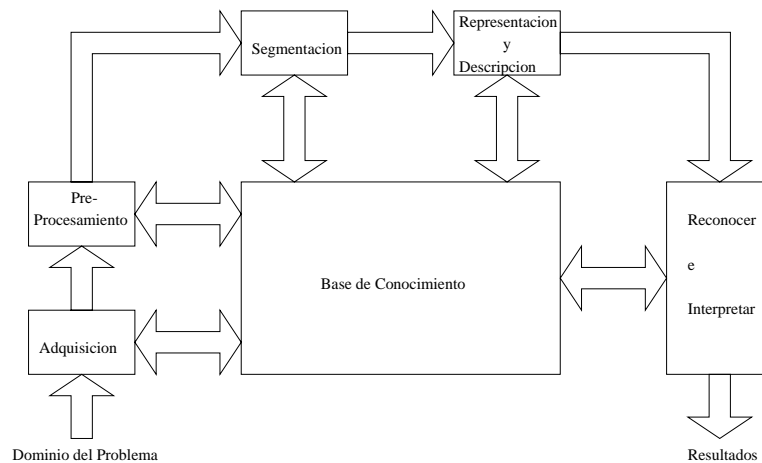


Figura 2.8: Pasos en el procesamiento de la imagen

muestran en la figura 12.8.2.

2.8. Ejercicios

1. Escribir un programa en C que implemente los diferentes modelos de emborronamiento descritos en el tema. El programa deberá ser el mismo para todos los modelos de emborronamiento y sólo variar en función de las opciones. Por ejemplo,

```
emborrona -i inputimage -a -s 5 -R 10 -o outputimage
```

realizaría un emborronamiento atmosférico con una desviación típica de 5 y un radio de 10 de la imagen inputimage y la escribiría en outputimage.

```
emborrona -i inputimage -m -e 3 -a 45 -o outputimage
```

realizaría un emborronamiento por movimiento en 3 píxeles con un ángulo de 45° de la imagen inputimage y la escribiría en outputimage. Crear la correspondiente página de manual (man page).

2. Con el mismo formato que en el programa anterior realizar la operación de añadir ruido para los diferentes modelos vistos en el tema. Crear la correspondiente página de manual (man page).

3. Escribir un programa usando cantata que simule los procesos de muestreo y cuantificación en una imagen. Es decir, que varíe la resolución y la cantidad de niveles de gris utilizados para obtener una imagen.

Capítulo 3

Sobre las imágenes y su almacenamiento en ficheros

3.1. Introducción

Una vez que una escena ha sido observada y capturada (o mejor dicho captada), por ejemplo mediante una fotografía, tenemos nuestra herramienta básica de trabajo, *una imagen*. Esta imagen podrá ser en blanco y negro (bivaluada), en niveles de gris o en color y a su vez podrá ser almacenada en un fichero o preparada para su transmisión usando un formato de imagen apropiado bien sea tiff, fits, gif, pbm, X11 bitmap, targa, etc.

En este tema analizaremos el proceso de obtención de imágenes tanto de niveles de gris como colores y comentaremos brevemente los diferentes formatos de ficheros de imágenes.

3.2. Fundamentos del Color

En 1666 Isaac Newton descubrió que cuando un rayo de luz solar pasa a través de un prisma de cristal el rayo de luz que sale no es blanco sino que está formado por un espectro continuo de colores que van desde el violeta al rojo. Podría decirse que el espectro de color puede dividirse en seis amplias regiones: violeta, azul, verde, amarillo, naranja y rojo. Sin embargo, cuando se ven todos los colores juntos ningún color del espectro termina bruscamente, más bien se mezcla suavemente con el siguiente (ver [12]).

La idea del color es inicialmente muy simple, los colores que los seres humanos perci-

bimos en un objeto vienen marcados por la naturaleza de la luz reflejada del objeto. Un cuerpo que refleja luz relativamente equilibrada en todas las longitudes de onda del visible aparece blanco. Un objeto que refleja en un rango limitado del espectro visible muestra algún tipo de sombra de color.

La caracterización de la luz es fundamental para la ciencia del color. Si la luz no tiene color su único atributo es la intensidad. Luz sin color es lo que vemos en las televisiones en blanco y negro.

Para una luz con color, el espectro electromagnético de luz visible va de 400 a 700 nm ($10^{-9}m$ aproximadamente). Se usan tres cantidades para describir una luz cromática: radiancia (un concepto que estudiamos en el capítulo anterior), luminancia y brillo. La luminancia se mide en lúmenes y es una medida de la cantidad de energía que el observador percibe. Por ejemplo, la luz emitida por una fuente que opera en el infrarrojo puede tener mucha energía, radiancia, pero no es visible por el observador y por tanto su luminancia es nula. Por último, el brillo es un descriptor subjetivo que se puede medir difícilmente, lleva asociado el concepto de luz sin color y es el responsable básico de la sensación de color.

Debido a la estructura del ojo humano, todos los colores se ven como una combinación de los tres llamados colores básicos Rojo(R), verde (G) y azul (B), usaremos la notación inglesa. Con la idea de producir un standard la Comisión Internacional de Iluminación (CIE) designó en 1931 las siguientes longitudes de onda para los tres colores primarios: azul= 435.8 nm, verde= 546.1 nm y rojo=700 nm, sin embargo es obvio que un sólo color no puede ser llamado rojo, verde a azul. Por tanto, tener tres longitudes de onda específicas para los tres colores no significa que estas tres componentes trabajando solas puedan generar todos los colores. Esto es importante, ya que se cree, erróneamente, que estos tres colores mezclados pueden producir todos los colores visibles. esto no es cierto salvo que también permitamos a las longitudes de onda que varien.

Si sumamos los colores primarios produciremos los colores secundarios, magenta (rojo más azul), cian (verde más azul) y amarillo (rojo más verde). Si mezclamos los tres colores primarios o un secundario con su color primario opuesto en las intensidades correctas tendremos luz blanca. Ver la figura 3.1 que también muestra los colores primarios y sus combinaciones para obtener los secundarios.

Diferenciar entre los colores primarios de luz y los colores primarios de pigmentación o colorantes es importante. En este caso, un color primario se define como uno que absorbe un color primario de luz y transmite los otros dos. En este caso pues, los colores

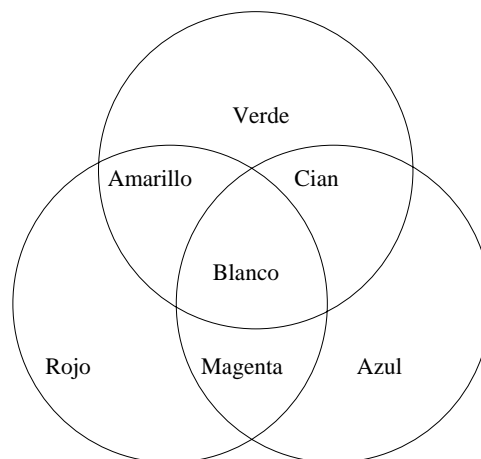


Figura 3.1: suma de colores primarios

básicos de pigmentación son magenta, cian y amarillo y los secundarios son rojo, verde y azul. Estos colores se muestran en la figura 3.2. Una combinación apropiada de las tres pigmentaciones primarias produce el negro.

La recepción de televisión en color es, como sabemos, un ejemplo de la naturaleza aditiva de los colores.

Las características que generalmente se usan para distinguir un color de otro son el brillo, el color (matiz) y la saturación. El brillo lleva asociado el concepto de intensidad. El matiz es un atributo asociado con la longitud de onda dominante en la mezcla de las ondas luminosas. La saturación se refiere a la cantidad de luz blanca mezclada con el matiz. Los colores puros del espectro están completamente saturados. Colores como el rosa están menos saturados siendo la saturación inversamente proporcional a la cantidad de luz blanca que se añade. Un ejercicio interesante es editar en el entorno de ventanas las propiedades de los mismos y ver como estos tres conceptos se relacionan.

3.3. Modelos de Color

El propósito de un modelo de color es facilitar la especificación de colores en algún formato standard. En esencia, un modelo de color es una especificación de un modelo de coordenadas 3-D y un subespacio dentro de ese sistema donde cada color se representa por un punto único.

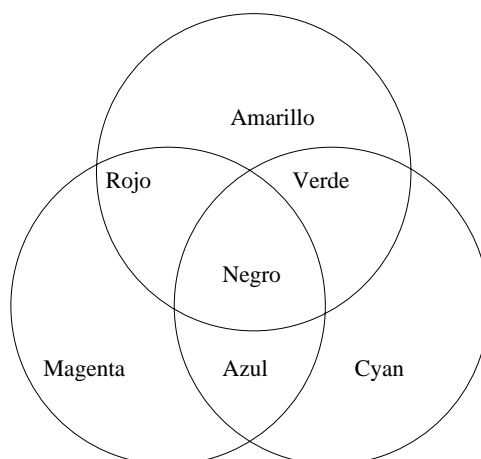


Figura 3.2: Colores secundarios

La mayoría de los modelos de color que se usan hoy se orientan hacia hardware como monitores o impresoras o aplicaciones de manipulación de color. El modelo orientado a hardware más común es el RGB (rojo-verde-azul), el modelo CMY (cian-magenta-amarillo) para impresoras en color y el YIQ que es el standard para la difusión de TV, en este caso la Y corresponde a luminancia y la I y Q son dos componentes cromáticas, ya las analizaremos. Para la manipulación del color se usan normalmente los modelos HSI (matiz, saturación e intensidad) y HSV (matiz, saturación y valor).

Los modelos de color más usados en el procesamiento de imágenes son RGB, YIQ y HSI. En la subsección siguiente detallaremos el modelo RGB, para un estudio más completo ver [12].

3.3.1. El modelo de color RGB

En el modelo RGB cada color aparece en sus componentes primarias espectrales rojo, verde y azul. El modelo está basado en un sistema de coordenadas cartesiano. El subespacio de interés es el cubo que se muestra en la figura 3.3. El gráfico es autoexplicativo.

Por conveniencia se supone que todos los colores han sido normalizados de forma que el cubo es unitario, es decir se supone que todos los valores de rojo, verde y azul están en el rango $[0, 1]$.

Las imágenes en el modelo de color RGB están formadas por tres planos de imágenes independientes, cada una de los colores primarios. Cuando son introducidas en un monitor

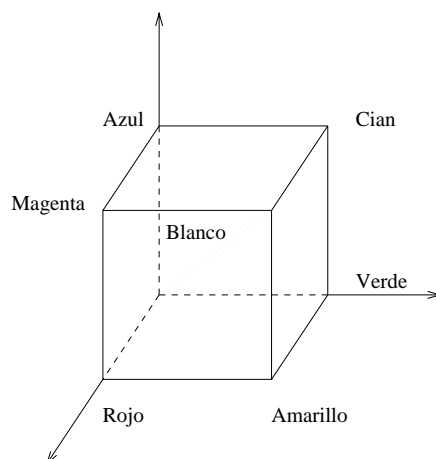


Figura 3.3: Cubo de color RGB. Los puntos en la diagonal principal tienen niveles de gris desde el negro en el origen al blanco en el punto $(1, 1, 1)$

RGB, las tres imágenes se combinan en la pantalla de fósforo para producir una imagen de color compuesta. Por tanto, el uso del modelo RGB para el procesamiento de imágenes tiene sentido cuando las imágenes vienen expresadas en términos de los tres planos de colores. Alternativamente, la mayoría de las cámaras en color que se usan para adquirir imágenes digitales utilizan el formato RGB, lo que hace si cabe más interesante este formato.

Uno de los mejores ejemplos de la utilidad del modelo RGB es el procesamiento de imágenes aéreas y de satélites multispectrales. Estas imágenes se obtienen para diferentes rangos espectrales. Por ejemplo, las imágenes LANDSAT se obtienen como mínimo en cuatro ventanas espectrales distintas de la misma escena. Dos ventanas están en el espectro visible y corresponden aproximadamente a verde y rojo; las otras dos están en el infrarrojo. Así pues cada plano de la imagen tiene sentido físico y la combinación de color utilizando el modelo RGB para procesamiento y visualización tiene sentido cuando se ve en una pantalla en color. La figura 3.4 muestra una imagen en color de verduras y frutas junto con sus tres bandas de color rojo, verde y azul. Obsérvese que el mayor nivel de gris en cada banda corresponde con el color predominante en cada objeto (fruta o verdura).

Sin embargo, no siempre es éste el mejor modelo de color para el procesamiento de imágenes. Consideremos, por ejemplo que queremos realzar una imagen en color de una cara humana, parte de la cual está oculta por sombra. Este problema puede ser abordado



(a)



(b)



(c)



(d)

Figura 3.4: (a) Imagen original en color. (b) Plano de rojo. Plano de verde y (c) plano de azul

como la igualación del histograma. Al tener tres bandas y puesto que el histograma trabaja sólo con las intensidades, cuando igualemos el histograma en cada banda, en cada una de ellas tendremos mejora, sin embargo, el resultado global, (cuando se combinen las tres bandas), será impredecible. Algunos otros modelos de color, por ejemplo el que discutiremos brevemente a continuación son más útiles para este tipo de problemas.

RGB es la elección más corriente para gráficos por ordenador ya que los CRT usan fósforo rojo, verde y azul para crear el color deseado. Sin embargo, RGB no es la forma más eficiente de describir las tres componentes de color. El igual ancho de banda da lugar a la misma profundidad del pixel y resolución del display para cada componente. Sin embargo, la sensibilidad de la componente de color del ojo humano es menor que la de la luminancia. Por estas razones, muchos métodos de codificación de imágenes y sistemas de transmisión utilizan la luminancia y dos diferencias de colores. Estos son los formatos YUV, YIQ, YCbCr y SMPTE 240M [44].

3.3.2. El modelo de color CMY

Como ya sabemos el cian, magenta y amarillo son los colores secundarios. Para convertir de colores primarios a secundarios hemos de realizar la siguiente transformación.

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

La figura 3.5(a,b,c) muestra las representaciones en las correspondientes bandas de la imagen en la figura 3.4(a). Para obtener estas imágenes se ha normalizado cada canal a uno independientemente, dividiéndolo por 255, a continuación se le ha restado a uno dicha cantidad y por último se ha multiplicado el resultado por 255.

3.3.3. El modelo de color YUV

Este es el formato de color se utiliza por los standards de TV NTSC, PAL y SECAM. La Y representa la componente de blanco y negro y la información de color U y V se añade para mostrar una imagen en color. La transformación es

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Observemos que $U = 0,492(B - Y)$ y que $V = 0,877(R - Y)$. Usualmente U y V son submuestreados por un factor de dos a cuatro en la dimensión espacial, ya que el ojo humano es mucho menos sensible a estos factores.. Esto es muy útil, por ejemplo, para reducir el número de bits utilizado en técnicas de compresión de imágenes.

3.3.4. El modelo de color YIQ

El modelo YIQ se usan en las emisiones de TV en color. Básicamente, YIQ es una recodificación de RGB para la eficiencia en la transmisión y para mantener la compatibilidad con los standards de TV monocromo. De hecho, la componente Y del sistema YIQ proporciona toda la información de video que se requiere para una televisión monocromo. La conversión de RGB a YIQ viene dada por

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,275 & -0,321 \\ 0,212 & -0,523 & 0,311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Este sistema fue diseñado teniendo en cuenta las características del sistema visual humano, en particular la mayor sensibilidad a los cambios en luminancia que a los cambios de matiz o saturación.. Así pues, este standard usa más bits (o ancho de banda) para representar Y y menos para I y Q.

Además de ser ampliamente usado, una de sus ventajas más importantes es que las informaciones de luminancia (Y) y color (I y Q) están separadas. la importancia de esta separación es que podemos procesar la luminancia sin afectar el color.

La figura 3.5(d,e,f) muestra las representaciones en las correspondientes bandas de la imagen en la figura 3.4(a).

Observemos que la figure 3.5(d) es una imagen en niveles de gris de la imagen original. Puesto que las bandas YUV tienen rangos distintos todas han sido llevadas al $[0,255]$ para la visualización. Tengamos en cuenta que tanto la banda U como V pueden tener valores negativos.

3.3.5. El modelo de color HSI

Como ya hemos visto el matiz es un atributo que describe la pureza de un color (puro amarillo, naranja o rojo), mientras que la saturación da una medida del grado en el que un color puro se diluye con luz blanca. La utilidad del modelo de color HSI se debe a dos

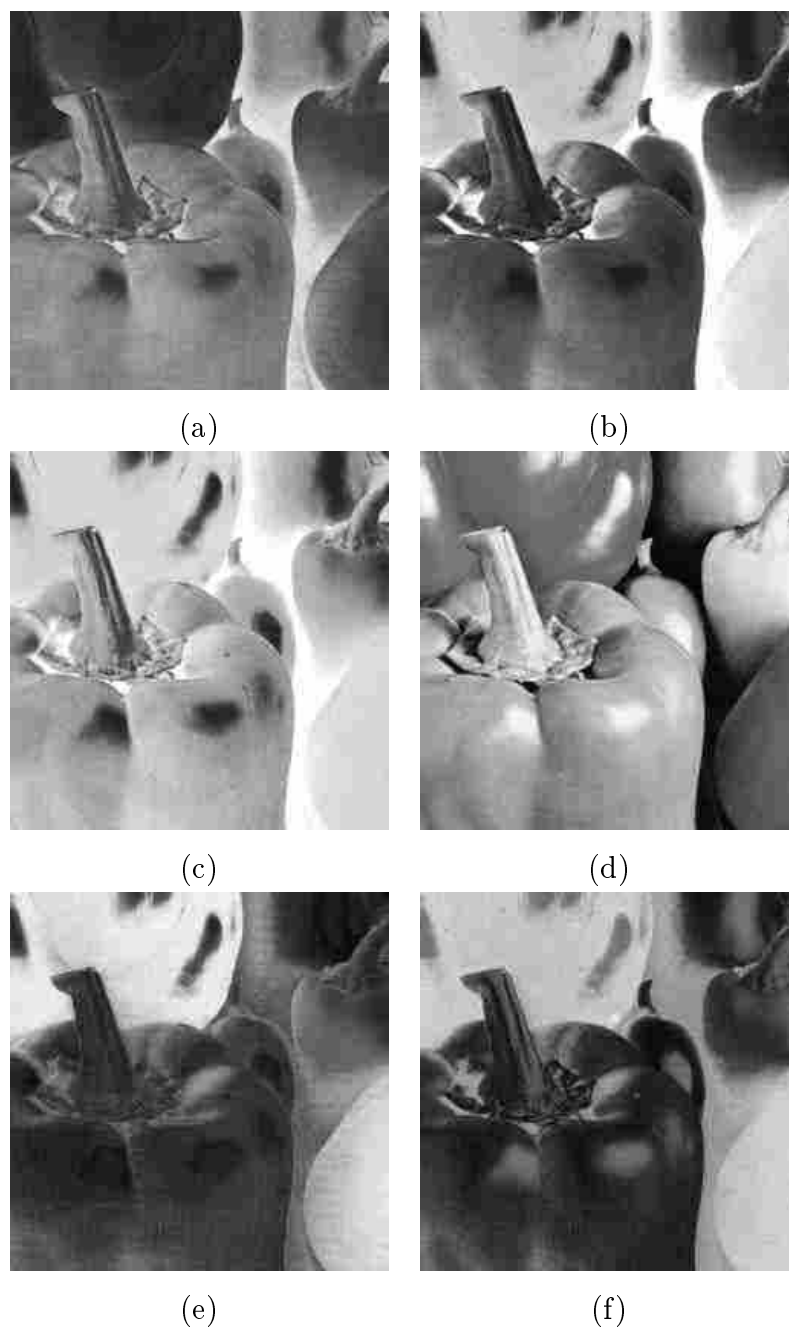


Figura 3.5: (a) Plano C del modelo CMY (b) Plano M del modelo CMY (c) Plano Y del modelo CMY (d) Plano Y del modelo YIQ (e) Plano I del modelo YIQ (f) Plano Q del modelo YIQ.

hechos principales. Por una parte, la componente de intensidad se separa de la información de color y además las otras dos componentes están muy relacionadas con la forma en la que el ser humano percibe el color. Estas características hacen este modelo muy apropiado para el desarrollo de algoritmos de procesamiento de imágenes basados en propiedades del sistema de visión humano.

Las fórmulas de conversión del modelo RGB a HSI y viceversa pueden encontrarse en [12].

3.4. Formatos de ficheros de imágenes

En esta sección utilizaremos el concepto de ficheros gráficos para describir ficheros que almacenan cualquier tipo de datos gráficos.

Los datos gráficos se dividen normalmente en dos clases: vectoriales y mapas de bits.

Los datos vectoriales se refieren normalmente a una forma de representar líneas, polígonos o curvas, y en general a objetos que se pueden dibujar fácilmente con líneas, especificando numéricamente puntos claves. Los mapas de bits están formados por un conjunto de valores numéricos que especifican los colores de los píxeles.

Estos son los dos tipos básicos de ficheros gráficos, los correspondientes a vectores y a mapas de bits. Sin embargo, hay otros tipos de formatos, los de descripción de escenas, animación, multimedia, hipertexto, hipermedia, 3D, VRML, audio y PDL entre otros. En esta sección describiremos solo y brevemente el formato de mapas de bits. El lector interesado puede consultar el libro [37] o la dirección <http://www.ora.com/gff> que contiene la versión on line de dicho libro así como referencias interesantes a los formatos de ficheros gráficos.

En esta sección discutiremos brevemente los conceptos asociados a ficheros de mapas de bits. Las componentes básicas de un fichero de mapas de bits son las siguientes: una cabecera y los datos como mapa de bits.

A veces, si toda la información no puede incluirse en la cabecera, el fichero puede contener un pie. También es posible que se almacene una paleta de colores, que suele estar después de la cabecera y si el fichero contiene varias imágenes, varias bandas, puede existir un mapa de bits que contenga las direcciones, como desplazamientos, donde están los mapas de bits de las imágenes.

Discutamos brevemente la cabecera. Su formato puede ser binario o ASCII y contiene

información sobre el mapa de bits que se encuentra en el fichero. Contendrá información sobre el tamaño de la imagen en filas y columnas, sobre un identificador que define el formato de almacenamiento, (gif, tiff, fits,..), etc.

Por último en el mapa de bits estará la imagen que puede estar almacenada por filas, por columnas o por planos de bits.

En general el formato de cada fichero de imágenes es diferentes pero todos ellos tienen básicamente la misma estructura que hemos procurado describir brevemente en esta sección.

3.5. Ejercicios

1. Utilizando `cantata` reproducir obtener las imágenes que aparecen en este capítulo.
2. Usar `cantata` para : convertir una imagen RGB a YIQ, muestrear promediando a la mitad de resolución las componentes IQ, a continuación pasar estas dos componentes al tamaño original de la imagen mediante la realización de un zoom de tamaño dos en cada dirección de la imagen. Pasar entonces esta nueva imagen YIQ a formato RGB y compararla con la original.

Capítulo 4

Transformaciones de Imágenes

En este capítulo analizaremos algunas transformadas discretas de señales e imágenes, así como algunas de sus aplicaciones. La teoría de las transformaciones ha jugado, y juega, un papel fundamental en el procesamiento de imágenes por su descripción alternativa de una imagen y por ser una herramienta para realizar operaciones como la resolución de ecuaciones lineales. Ambos conceptos serán tratados en este capítulo. Es muy importante tener en cuenta la enorme aplicación de las transformaciones de imágenes en campos como la restauración, codificación, mejora, segmentación, etc.

Aunque el material de este capítulo trata fundamentalmente del desarrollo de transformaciones bidimensionales y sus aplicaciones, algunos conceptos serán introducidos a partir de funciones unidimensionales.

En este tema no vamos a estudiar importantes transformaciones de imágenes como las transformaciones basadas en wavelets o los espacios de escala, conceptos que han adquirido un enorme auge en los últimos años y que serán tratados, si es posible, en temas posteriores.

4.1. Transformada de Fourier. Caso Continuo.

4.1.1. Transformada de Fourier Unidimensional

Sea $f(x)$ una función continua de variable real, recordemos que nuestro objetivo es buscar una representación alternativa de esa función. La transformada de Fourier de $f(x)$,

que notaremos $F(u)$ se define mediante

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp[-2\pi i u x] dx \quad (4.1)$$

Dado $F(u)$, $f(x)$ puede obtenerse por la transformada de Fourier inversa

$$f(x) = \int_{-\infty}^{\infty} F(u) \exp[2\pi i u x] du \quad (4.2)$$

La transformada de Fourier puede no existir, salvo que f cumpla algunas condiciones. Condiciones suficientes son:

1. $\int_{-\infty}^{\infty} |f(x)| dx < \infty$,
2. $f(x)$ debe tener sólo un número finito de discontinuidades y un número finito de máximos y mínimos en cualquier rectángulo finito,
3. $f(x)$ no puede tener discontinuidades infinitas.

ver ([46]) y referencias que allí se citan.

En este tema estudiaremos casi siempre funciones $f(x)$ que son reales. Sin embargo, la transformada de Fourier de una función real en general será compleja; es decir

$$F(u) = R(u) + iI(u)$$

donde $R(u)$ e $I(u)$ son respectivamente las partes real e imaginaria de $F(u)$ con $i = \sqrt{-1}$. La ecuación anterior puede escribirse en forma exponencial

$$F(u) = |F(u)| \exp[i\phi(u)]$$

donde

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

y

$$\phi(u) = \arctg[I(u)/R(u)]$$

La función de magnitud $|F(u)|$ recibe el nombre de espectro de Fourier de $f(x)$ y $\phi(u)$ recibe el nombre de fase. El cuadrado del espectro suele recibir el nombre de espectro de potencias.

La variable u suele recibir el nombre de frecuencia. Este nombre proviene de que usando la fórmula de Euler

$$\exp[2\pi i u x] = \cos(2\pi u x) + i \operatorname{sen}(2\pi u x)$$

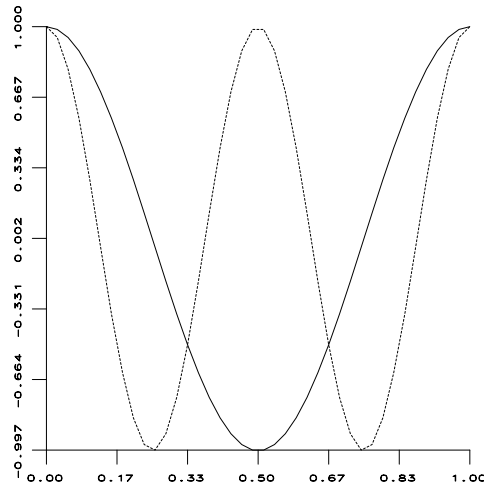


Figura 4.1: Representación de $\cos(2\pi x)$, línea continua, y $\cos(2\pi 2x)$, línea discontinua que corresponden a uno o dos ciclos por unidad de tiempo.

y por tanto u representa el número de ciclos completos por unidad de x .

La figura 4.1 representa dos ondas cosenos, una con un ciclo en el intervalo $[0, 1]$ y otra con dos ciclos.

Ejemplo. 1 Consideremos la función $f_\epsilon(x)$ definida por

$$f_\epsilon(x) = \begin{cases} 1/\epsilon & \text{si } |x| < \epsilon/2 \\ 0 & \text{en caso contrario} \end{cases}$$

Su transformada de Fourier se obtiene mediante

$$F_\epsilon(u) = \int_{-\epsilon/2}^{\epsilon/2} \frac{1}{\epsilon} \exp[-2\pi i u x] dx = \frac{\text{sen}(\pi u \epsilon)}{\pi u \epsilon}$$

que se muestra en la figura 4.2, su espectro vale

$$|F_\epsilon(u)| = \left| \frac{\text{sen}(\pi u \epsilon)}{\pi u \epsilon} \right|$$

Como vemos su transformada de Fourier es real. Sin embargo, si consideramos la función

$$h_\epsilon(x) = f_\epsilon(x - x_0)$$

puede comprobarse que su transformada de Fourier es compleja, pero que el espectro es el mismo.

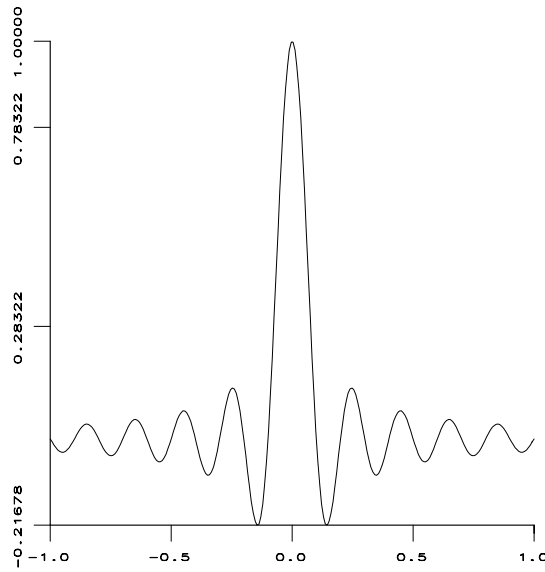


Figura 4.2: Función $\frac{\text{sen}(\pi 10x)}{\pi 10x}$

Es importante recordar que las funciones $f_\epsilon(x)$ que hemos definido dan lugar a la delta de Dirac que se define

$$\delta(x) = \lim_{\epsilon \rightarrow 0} f_\epsilon(x)$$

que cumple

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

y para cualquier función $g(x)$

$$\int_{-\infty}^{\infty} \delta(x) g(x) dx = g(0)$$

es importante entender esta función. Además, las funciones de la forma $\delta(x - x_0)$ con $x_0 \in \mathcal{R}$ de ella derivadas cumplen

$$\int_{-\infty}^{\infty} \delta(x - x_0) dx = 1$$

y para cualquier función $g(x)$

$$\int_{-\infty}^{\infty} \delta(x - x_0) g(x) dx = g(x_0)$$

podemos, por tanto, decir que $\delta(x - x_0)$ está localizada en x_0 y que la fuerza del impulso viene determinada por el valor de $f(x)$ en $x = x_0$. Es usual representar los impulsos como una flecha en x_0 con una altura igual a la fuerza del impulso, es decir con una altura de $f(x_0)$.

4.1.2. Interpretación física de la transformada de Fourier unidimensional

La fórmula 4.2 puede usarse para dar una interpretación física a $F(u)$. Examinemos primero la función $\exp[2\pi i u x]$. Sus partes real e imaginaria son $\cos(2\pi u x)$ y $\sin(2\pi u x)$. Consideremos la función $\cos(2\pi u x)$. Esta función tiene u ciclos por unidad de distancia en x (por ello, como dijimos, se le llama frecuencia a u). Además. Su periodicidad espacial es $1/|u|$. Cambiando u podemos generar patrones con periodicidad espacial de 0 a ∞ .

Por tanto, la fórmula 4.2 puede interpretarse como que $f(x)$ puede considerarse como una combinación lineal de patrones periódicos de la forma $\exp[2\pi i u x]$. Evidentemente $F(u)$ es un factor de peso que mide la contribución de cada patrón.

4.1.3. Transformada de Fourier Bidimensional

La transformada de Fourier puede extenderse fácilmente a dimensión dos y superior. La definición en dimensión dos es la siguiente. El par de transformadas de Fourier queda definido, en los casos en los que exista, mediante

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-2\pi i (ux + vy)] dx dy \quad (4.3)$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[2\pi i (ux + vy)] du dv \quad (4.4)$$

Ejemplo. 2 *Es importante que como ejercicio se haga la extensión al caso bidimensional de la función delta de Dirac y se calcule su espectro.*

En este caso la delta de Dirac cumple

$$\int_{-\infty}^{\infty} \delta(x, y) dx dy = 1$$

y para cualquier función $g(x, y)$

$$\int_{-\infty}^{\infty} \delta(x, y) g(x, y) dx dy = g(0, 0)$$

es importante esta función y las de la forma $\delta(x - x_0, y - y_0)$ con $x_0, y_0 \in \mathcal{R}$ de ella derivadas.

Ejercicio. 1 Calcular la transformada de Fourier de

$$\cos(2\pi(\alpha x + \beta y))$$

Ejercicio. 2 Probar que

$$\int \int \exp[-2\pi i(ux + vy)] dudv = \delta(x, y). \quad (4.5)$$

Demostrar intuitivamente, la demostración formal puede encontrarse en [18] y requiere los conceptos de factores de convergencia, que

$$\sum_{k=-\infty}^{k=\infty} \exp[2\pi iku] = \sum_{k=-\infty}^{k=\infty} \delta(u - k) \quad (4.6)$$

y en general que

$$\sum_{k=-\infty}^{k=\infty} \exp[2\pi iku\Delta] = \frac{1}{\Delta} \sum_{k=-\infty}^{k=\infty} \delta(u - k/\Delta) \quad (4.7)$$

Probar entonces que la transformada de Fourier de

$$\text{comb}(x, y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - m, y - n)$$

es

$$\text{comb}(u, v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(u - m, v - n)$$

para concluir que la transformada de Fourier de

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x - m\Delta x, y - n\Delta y)$$

es

$$\frac{1}{\Delta x} \frac{1}{\Delta y} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(u - m/\Delta x, v - n/\Delta y)$$

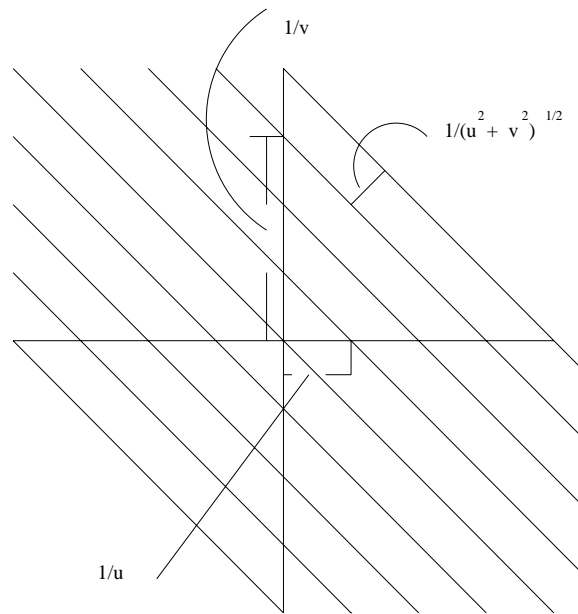


Figura 4.3: Líneas correspondientes a máximos de la función $\cos(2\pi(u x + v y))$

4.1.4. Interpretación física de la transformada de Fourier bidimensional

La ecuación (4.4) puede usarse para dar una interpretación física de $F(u, v)$ y de las coordenadas u y v , supuesto que x e y representan coordenadas espaciales.

Examinemos primero la función $\exp[2\pi i(u x + v y)]$. Las partes real e imaginaria son $\cos[2\pi(u x + v y)]$ y $\sin[2\pi(u x + v y)]$, respectivamente. En la figura 4.3 mostramos las líneas correspondientes a los máximos de la función $\cos[2\pi(u x + v y)]$. Es claro que si tomamos una sección de este patrón bidimensional paralelo al eje x , tenemos u ciclos por unidad de distancia. Mientras que en el eje y tendremos v ciclos. Es por ello por lo que u y v reciben el nombre de frecuencias espaciales. El periodo espacial del patrón es $(u^2 + v^2)^{-1/2}$.

De la discusión anterior es claro que $\exp[2\pi i(u x + v y)]$ es un patrón bidimensional. Cambiando u y v podemos generar patrones con periodos espaciales que van de 0 a ∞ en cualquier dirección del plano.

La ecuación 4.4 puede interpretarse, por tanto, como que $f(x, y)$ es una combinación lineal de patrones periódicos elementales de la forma $\exp[2\pi i(u x + v y)]$. Evidentemente, $F(u, v)$ es sólo un factor de peso que mide la contribución relativa de los patrones

elementales.

4.1.5. Propiedades de la Transformada de Fourier bidimensional

El espectro de muchas imágenes decrece de forma muy rápida cuando aumenta la frecuencia y por tanto los términos de frecuencias altas tienden a hacerse oscuros muy pronto al mostrarse en una imagen o en una señal unidimensional. Una técnica usual es mostrar la imagen usando

$$D(u, v) = \log(1 + |F(u, v)|)$$

u otros reescalados.

Veamos a continuación una serie de propiedades de la transformada de Fourier cuya demostración se deja como ejercicio.

Notemos por $\mathcal{F}(f)$ la transformada de Fourier de la función $f(x, y)$, en otras palabras $\mathcal{F}(f(x, y)) = F(u, v)$.

1. Linealidad.

$$\mathcal{F}(af_1(x, y) + bf_2(x, y)) = a\mathcal{F}(f_1(x, y)) + b\mathcal{F}(f_2(x, y))$$

2. Escalado.

$$\mathcal{F}(f(\alpha x, \beta y)) = \frac{1}{|\alpha\beta|} F(u/\alpha, v/\beta)$$

3. Propiedad de Traslación.

$$\mathcal{F}(f(x - \alpha, y - \beta)) = F(u, v) \exp[-2\pi(u\alpha + v\beta)]$$

4. Teorema de Parseval

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(x, y) f_2^*(x, y) dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F_1(u, v) F_2^*(u, v) du dv$$

donde * denota complejo conjugado.

4.1.6. Convolución y Correlación

En esta sección consideraremos dos relaciones entre transformadas de Fourier que constituyen la unión básica entre los dominios espacial y de las frecuencias. Estas relaciones,

llamadas convolución y correlación, son de fundamental importancia para entender el procesamiento de imágenes usando transformada de Fourier. Para clarificar los conceptos que se usan, empezaremos la discusión considerando la convolución en el caso unidimensional para después hacer lo mismo con la correlación.

Convolución

La convolución de dos funciones $f(x)$ y $g(x)$ que se nota $(f * g)(x)$ y se define mediante la integral

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha$$

donde α es la variable de integración. Veamos la idea gráficamente.

Ejemplo. 3 *La figura 4.4 muestra la convolución de dos funciones $f(x)$ y $g(x)$ en todos sus pasos. Puede probarse que*

$$(f * g)(x) = \begin{cases} x/2 & 0 \leq x \leq 1 \\ 1 - x/2 & 1 \leq x \leq 2 \\ 0 & \text{en caso contrario} \end{cases}$$

Es muy interesante estudiar este mismo ejemplo pero suponiendo que tanto la función $f(x)$ como $g(x)$ son periódica de periodo T y entonces examinar la forma que tiene la convolución.

Ejemplo. 4 *El segundo ejemplo nos acerca a las problemas de muestreo discreto que están, como veremos, siempre relacionados con la función $\delta(x)$ antes definida. Supongamos que la función $f(x)$ definida en la figura 4.5 se convoluciona con la función $g(x) = \delta(x + T) + \delta(x) + \delta(x - T)$. Es fácil probar que la convolución simplemente copia $f(x)$ en la localización de cada impulso, ver figura 4.5.*

La importancia de la convolución en el dominio de las frecuencias viene del hecho de que la transformada de Fourier de $(f * g)(x)$ es $F(u)G(u)$ donde $F(u)$ y $G(u)$ son las transformadas de Fourier de $f(x)$ y $g(x)$ respectivamente.

Veamos ahora la definición en el caso bidimensional

$$(f * g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)g(x - \alpha, y - \beta)d\alpha d\beta$$

En este caso la transformada de Fourier de $(f * g)(x, y)$ es, como ya hemos dicho, $F(u, v)G(u, v)$. Una extensión obvia del caso unidimensional.

La convolución es ilustrada gráficamente para el caso 2-D en la figura 4.6

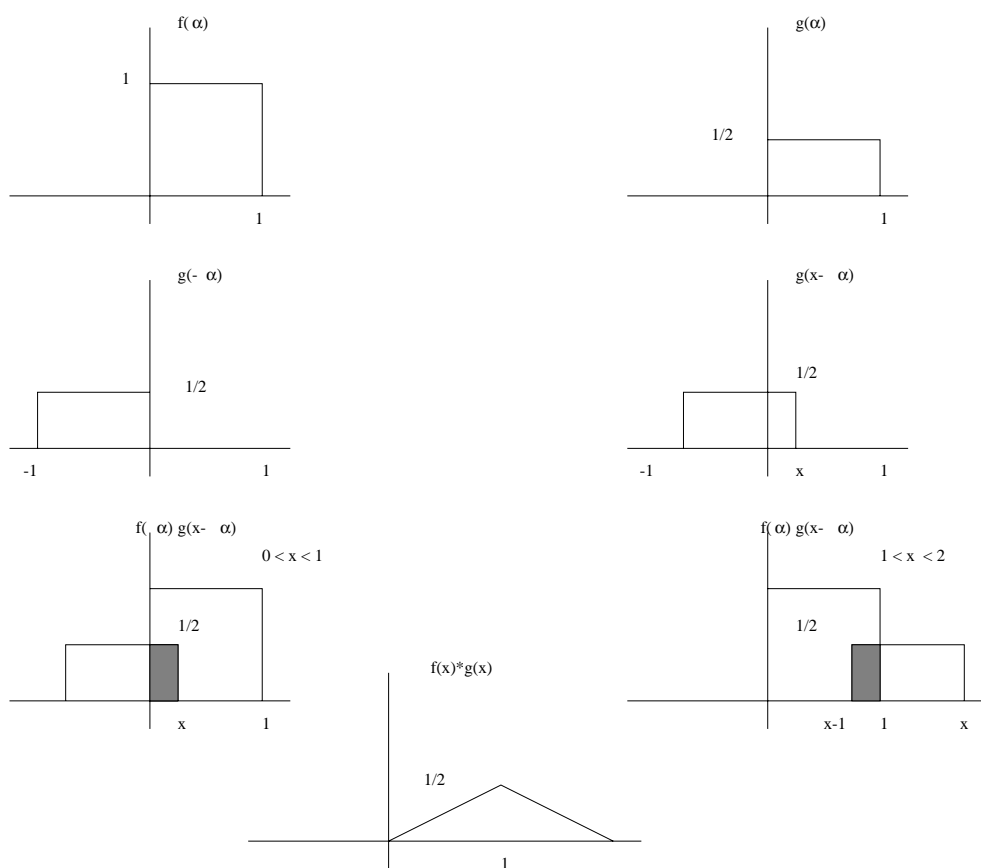


Figura 4.4: Convolución Unidimensional.

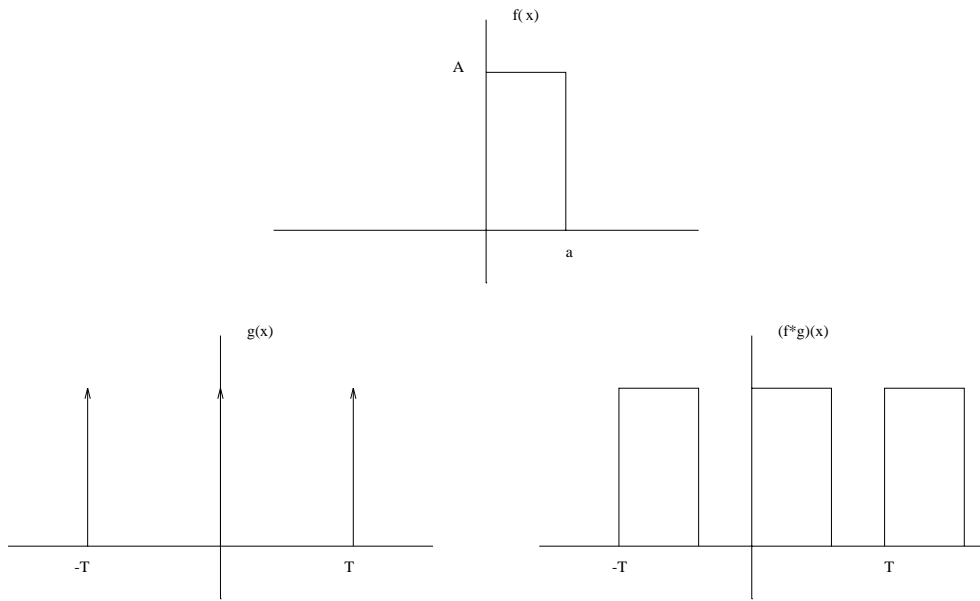


Figura 4.5: Ejemplo de convolución.

Correlación

La correlación cruzada entre dos funciones f y g se define mediante

$$(f \circ g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f^*(\alpha, \beta) g(x + \alpha, y + \beta) d\alpha d\beta$$

puede probarse fácilmente que

$$(f \circ g)(x, y) = (g \circ f)^*(-x, -y)$$

Para la correlación se verifica que $\mathcal{F}(f \circ g) = [\mathcal{F}(f)]^* \mathcal{F}g$. Por último, para pasar al muestreo discreto, se deja como ejercicio estudiar la relación entre convolución y correlación.

4.2. Señal continua y muestreo discreto. Reconstrucción de la señal continua a partir del muestreo

El procesamiento de imágenes se basa en un muestreo discreto de la imagen original. Es obvio que uno se pregunta bajo qué condiciones el muestreo discreto puede recuperar la imagen original. Las condiciones las establece el teorema que Whittaker-Kotelnikov-Shannon que veremos en el caso unidimensional.

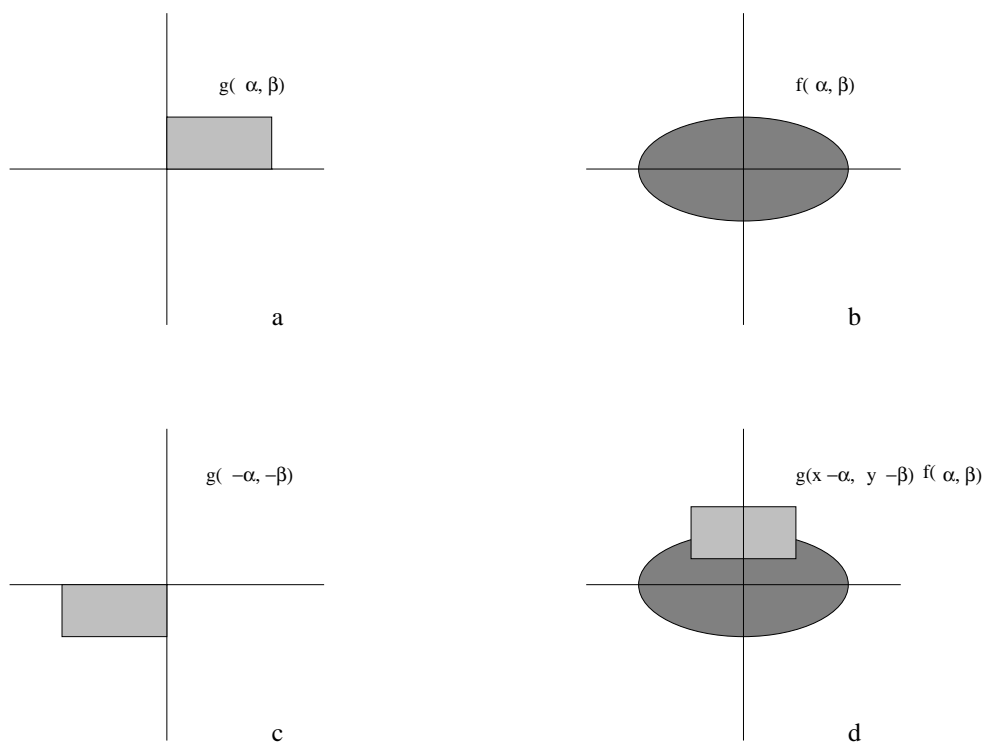


Figura 4.6: Representación gráfica de la convolución en dos dimensiones. Las dos funciones a convolucionar aparecen en (a) y (b). En (c) la primera función se rota 180° y (d) muestra el producto $g(x-\alpha, y-\beta)f(\alpha, \beta)$. El volumen bajo la función producto es la evaluación de la convolución para el valor de (x, y) elegido.

4.2. SEÑAL CONTINUA Y MUESTREO DISCRETO. RECONSTRUCCIÓN DE LA SEÑAL CONTINUA

Consideremos una función unidimensional $f(t)$ que deseamos representar mediante muestras $f(k\Delta)$ donde k toma valores enteros en $-\infty$ a ∞ y Δ es el periodo de muestreo.

Lo que queremos hacer es encontrar una función de interpolación $g(t)$ tal que

$$f(t) = \sum_{k=-\infty}^{\infty} f(k\Delta)g(t - k\Delta) \quad (4.8)$$

supondremos que f y g tienen transformada de Fourier.

Desarrollemos el proceso intuitivo y formal al mismo tiempo. Supongamos que $f(t)$ está definida sobre la recta real y que su transformada de Fourier es nula fuera del intervalo $[-W, W]$. Esta idea se muestra en la figura 4.7(a,b). Una función que cumple esta condición se dice de banda limitada. Muestrear consiste, como sabemos, en multiplicar por un tren de pulsos a distancia Δx como muestra la figura 4.7(c) cuya transformada de Fourier se muestra en figura 4.7(d). Por el teorema de convolución, multiplicación en el dominio espacial corresponde a convolución en el dominio de las frecuencias. La transformada de Fourier de la imagen discreta se muestra en la figura 4.7(f). Observemos que la transformada de Fourier es periódica con periodo $1/\Delta x$ y las repeticiones pueden solaparse como ocurre en la figura 4.7(f). Sin embargo, si

$$\Delta x \leq 1/(2W)$$

no existe solapamiento como muestra la figura 4.7(g,h). Observemos que todavía no hemos recuperado la señal original pues tenemos repeticiones periódicas del espectro original. Si ahora multiplicamos por

$$G(u) = \begin{cases} 1 & -W \leq u \leq W \\ 0 & \text{en caso contrario} \end{cases}$$

volvemos a aislar $F(u)$ completamente. veamos el proceso formal

Retomemos la ecuación 4.8. Teniendo en cuenta que

$$f(k\Delta)g(t - k\Delta) = \int f(\tau)g(t - \tau)\delta(\tau - k\Delta)d\tau$$

lo que deseamos es, como ya hemos dicho, que

$$f(t) = \sum_{k=-\infty}^{\infty} f(k\Delta)g(t - k\Delta)$$

que es lo mismo que decir que

$$f(t) = \int g(t - \tau)[f(\tau) \sum_k \delta(\tau - k\Delta)]d\tau$$

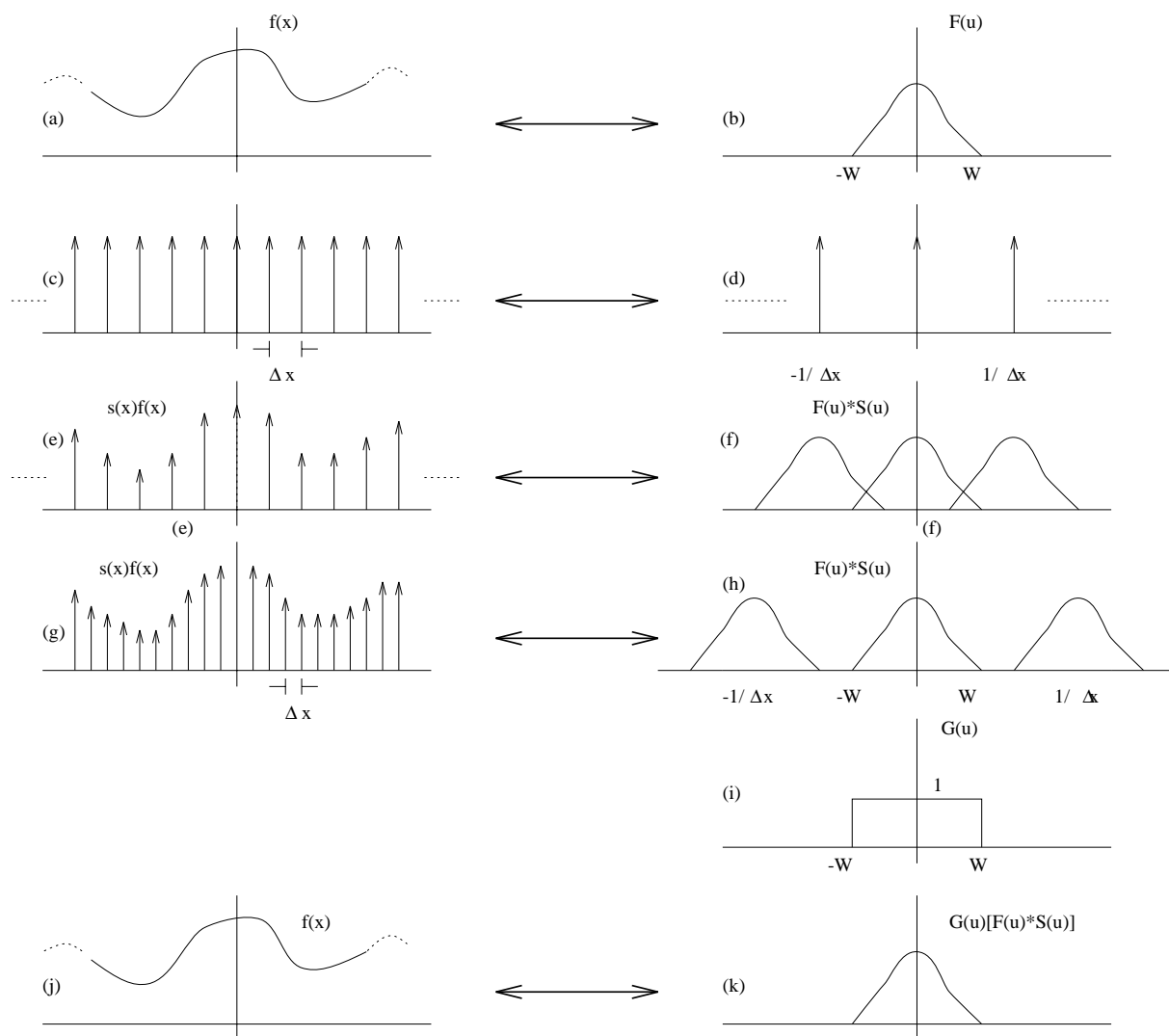


Figura 4.7: Muestreo

4.2. SEÑAL CONTINUA Y MUESTREO DISCRETO. RECONSTRUCCIÓN DE LA SEÑAL CONTINUA

ahora bien, la transformada de Fourier de $f_s(t) = f(t) \sum_k \delta(t - k\Delta)$ es

$$F_s(u) = \frac{1}{\Delta} \int_{-\infty}^{\infty} F(\tau) \sum_k \delta(u - k/\Delta - \tau) d\tau = \frac{1}{\Delta} \sum_k F(u - \frac{k}{\Delta})$$

y por tanto la transformada de Fourier de

$$\int g(t - \tau)(f(\tau) \sum_k \delta(\tau - kT)) d\tau$$

será

$$\frac{G(u)}{\Delta} \sum_k F(u - \frac{k}{\Delta})$$

Veamos entonces que condiciones son suficientes para que se cumpla

$$F(u) = \frac{G(u)}{\Delta} \sum_k F(u - \frac{k}{\Delta}) \quad (4.9)$$

Supongamos que $F(u) = 0$ para $|u| > W$ entonces si $1/\Delta > 2W$ o bien $\Delta < 1/(2W)$ nos garantizamos que (4.9) se cumple en la región donde $F(u)$ es no nula tomando

$$G(u) = \begin{cases} \Delta & |u| < W \\ 0 & \text{en caso contrario} \end{cases}$$

El proceso es pues un producto en el dominio de Fourier y por tanto una convolución en el espacio del tiempo. Es decir, ya que

$$\frac{1}{\Delta} \sum_k F(u - \frac{k}{\Delta}) \quad \text{y} \quad f_s(t) = f(t) \sum_k \delta(t - k\Delta)$$

son transformadas de Fourier inversas y que

$$G(u) \quad \text{y} \quad \Delta \frac{\text{sen}(2\pi Wt)}{\pi Wt}$$

también lo son, tendremos

$$f(t) = \int f(\alpha) \sum_k \delta(\alpha - k\Delta) \Delta \frac{\text{sen}(2\pi W(t - \alpha))}{\pi W(t - \alpha)} d\alpha = \Delta \sum_{n=-\infty}^{n=\infty} f(n\Delta) \frac{\text{sen}(2\pi W(t - n\Delta))}{\pi(t - n\Delta)}$$

Ejemplo. 5 Estudiar como muestrear la función $2 \cos(2\pi(3x + 4y))$ para que pueda ser reconstruida a partir del muestreo.

Ejemplo. 6 Si muestreamos la función $\cos(2\pi x) + \cos(2\pi 4x)$ con un espaciado $\Delta = 0,5$ e intentamos reconstruir la señal original a partir de sus muestras qué obtenemos. ?
Qué espaciado deberíamos usar para reconstruir la señal original ?

4.3. Series de Fourier.

Consideremos una función $f(t)$ definida sobre la recta real y consideremos ahora, de nuevo, la función

$$f_s(t) = f(t) \sum_k \delta(t - k\Delta)$$

esta función tiene como transformada de Fourier

$$F_s(u) = \sum_k f(k\Delta) \exp[-2\pi k\Delta u] = \sum_n F(u - n\Delta)/\Delta$$

que recibe el nombre de serie de Fourier. Observemos además que la función $F_s(u)$ es periódica de periodo $1/\Delta$.

Supongamos que ahora restringimos $F_s(u)$ al intervalo $[-1/(2\Delta), 1/(2\Delta)]$ haciendolo cero fuera de él, es decir,

$$\hat{F}(u) = \begin{cases} F_s(u) & |u| < 1/(2\Delta) \\ 0 & \text{en caso contrario} \end{cases}$$

entonces

$$\hat{f}(x) = \int_{-1/(2\Delta)}^{1/(2\Delta)} F_s(u) \exp[2\pi iux] du$$

en particular estamos interesados en conocer qué ocurre en los puntos de la forma $k\Delta$, tendremos

$$\hat{f}(k\Delta) = \int_{-1/(2\Delta)}^{1/(2\Delta)} F_s(u) \exp[2\pi iuk\Delta] du \quad (4.10)$$

$$= \int_{-1/(2\Delta)}^{1/(2\Delta)} \left[\sum_n f(n\Delta) \exp[-2\pi(n-k)\Delta u] \right] du \quad (4.11)$$

$$= \sum_n f(n\Delta) \frac{\text{sen}(\pi(n-k)\Delta)}{\pi(n-k)\Delta} \quad (4.12)$$

$$= f(k\Delta) \quad (4.13)$$

Es decir, calculando la transformada inversa a $\hat{F}(u)$ recuperamos la señal original en los puntos que hemos muestreado, lo cual no nos garantiza que la recuperemos en todos los puntos.

4.4. Recuperación de la señal original a partir del muestreo en un número finito de puntos

Los resultados sobre muestreo que vimos en la sección 4.2 se aplican a funciones que son de duración ilimitada en el tiempo. Esto implica un intervalo de muestreo infinito en el espacio del tiempo. Sin embargo, nuestra capacidad de muestreo es finita. La situación se muestra en la figura 4.8(a-f), suponiendo que no hay solapamiento, (aliasing). Un muestreo en el intervalo $[0, X]$ puede ser representado como el resultado de multiplicar figura 4.8(e) por la función

$$h(x) = \begin{cases} 1 & |x| \leq X/2 \\ 0 & \text{en caso contrario} \end{cases}$$

esta función recibe el nombre de ventana y se muestra con su transformada de Fourier en figura 4.8(g,h). Los resultados de la multiplicación se muestran en figura 4.8(i-j). Observamos que como $H(u)$ tiene componentes que se extienden hasta infinito, la convolución introduce una distorsión en el dominio de las frecuencias de la función que ha sido muestreada y limitada a una región finita por $h(x)$ como se muestra en figura 4.8(j). Así pues, incluso si la separación entre las muestras cumple el teorema de muestreo para la imagen original, la introducción de $h(x)$ hace la recuperación completa de la función a partir de un muestreo finito es, en general, imposible (por culpa de $h(x)$ que tiene una transformada de Fourier que no es de banda limitada).

El problema se complica por tanto. Sin embargo, sí es posible llevar a cabo la recuperación de la señal continua original a través de un muestreo finito cuando

1. $f(x)$ es periódica de periodo X y es de banda limitada (observemos que toda función periódica de periodo X no es necesariamente de banda limitada, por ejemplo $\sum_n \delta(x - nX)$)
2. El espaciado entre las observaciones cumple el teorema de Whittaker-Shannon.

Ejemplo. 7 *Estas ideas se pueden entender muy fácilmente si usamos como ejemplo la función $\cos(2\pi x)$. Primero, la muestreamos en el intervalo $[-1/2, 1/2]$. Segundo, tenemos en cuenta que su banda está limitada a $[-1, 1]$, recordemos que su transformada de Fourier es $(\delta(-1) + \delta(1))/2$. Entonces por el teorema de Whittaker-Shannon deberemos muestrear con un Δ de $1/2$ como máximo, es decir tres puntos como mínimo.*

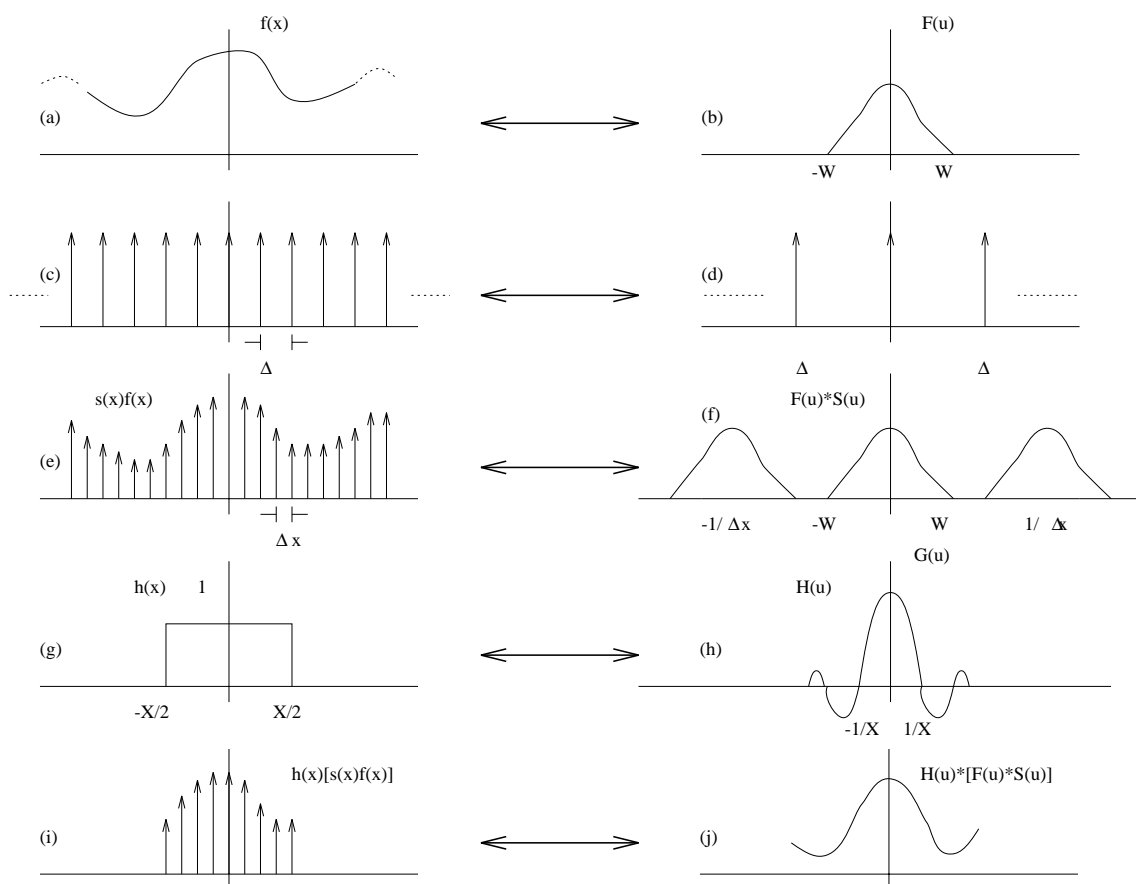


Figura 4.8: Muestreo de duración finita

4.5. Desde la transformada de Fourier Continua hacia la discreta

Veamos como introducir la transformada de Fourier discreta a partir de lo que sabemos de la continua.

Supongamos que nuestra imagen es muestreada en los puntos $0, T, 2T, \dots, (N-1)T$, (ya hemos estudiado las condiciones que debe cumplir T para recuperar la señal original si ésta es de banda limitada). Tenemos, por tanto, la función

$$g(t) = f(t) \sum_{k=0}^{N-1} \delta(t - kT)$$

Su transformada de Fourier es continua y vale

$$G(u) = \sum_{n=0}^{N-1} f(nT) \exp[-2\pi i n T u]$$

Observemos que:

1. $G(u)$ es periódica de periodo $1/T$,
2. si se dan las condiciones apropiadas en cuanto al ancho de banda de $f(x)$, es decir si sus frecuencias están en $[0, 1/T]$, tendremos $G(k/(NT)) = F(k/(NT))$, para $k = 0, 1, \dots, (N-1)$, ver ejercicio 4.

Sea

$$\hat{G}(u) = \begin{cases} \frac{1}{N} G(u) & \text{si } u \in [0, 1/T] \\ 0 & \text{en caso contrario} \end{cases}$$

muestreémosla en los puntos $k/(NT)$ con $k = 0, 1, \dots, (N-1)$ y definamos

$$\hat{g}(x) = \sum_{k=0}^{N-1} \hat{G}(k/(NT)) \exp[2\pi i x k/(NT)]$$

Si la muestreamos en los puntos lT tenemos

$$\hat{g}(lT) = \sum_{k=0}^{N-1} \hat{G}(k/(NT)) \exp[2\pi i l k/N]$$

además, utilizando las propiedades de ortogonalidad de $\exp[2\pi i l k/N]$ puede probarse fácilmente que

$$\hat{G}(k/(NT)) = \frac{1}{N} \sum_{l=0}^{N-1} \hat{g}(lT) \exp[2\pi i l k/N]$$

cumpléndose además que

$$\hat{g}(lT) = \sum_{k=0}^{N-1} \hat{G}(k/(NT)) \exp[2\pi ilk/N] \quad (4.14)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} f(nT) \exp 2\pi ink/N \exp[-2\pi ilk/N] \quad (4.15)$$

$$= f(lT) \quad (4.16)$$

Ver [8] para detalles.

Por tanto, podemos escribir,

$$f(lT) = \sum_{k=0}^{N-1} \hat{G}(k/(NT)) \exp[2\pi ilk/N]$$

y

$$\hat{G}(k/(NT)) = \frac{1}{N} \sum_{l=0}^{N-1} f(lT) \exp[2\pi ilk/N]$$

siendo además $\hat{G}(k/(NT)) = \frac{1}{N} F(k/(NT))$ si se cumplen las condiciones de muestreo. A las dos ecuaciones anteriores se les llama par de transformadas de Fourier discretas.

4.6. Transformada de Fourier Discreta. Caso Bidimensional.

El material de este apartado trata fundamentalmente del uso de la transformada de Fourier bidimensional como representación alternativa de una imagen y como herramienta para calcular convoluciones y correlaciones. Su relación con las matrices circulantes por bloques se estudiará detenidamente en la sección siguiente.

Supongamos que una matriz bidimensional, correspondiente a una imagen, \mathbf{f} , es escrita como

$$\mathbf{f} = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{pmatrix}$$

Una transformación general discreta de la imagen \mathbf{f} de orden $M \times N$ produce también una matriz \mathbf{F} de orden $M \times N$ dada por

$$\mathbf{F} = \mathbf{P} \mathbf{f} \mathbf{Q}$$

donde \mathbf{P} y \mathbf{Q} son matrices no singulares de orden $M \times M$ y $N \times N$ respectivamente.

Es muy importante tener en cuenta que no hemos especificado nada sobre las matrices \mathbf{P} y \mathbf{Q} , por lo que no hemos asignado nombre a la transformación. Observemos además que

$$F(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} P(u, m) f(m, n) Q(n, v)$$

Al ser \mathbf{P} y \mathbf{Q} matrices no singulares, sus inversas están definidas de modo único y podemos escribir

$$\mathbf{f} = \mathbf{P}^{-1} \mathbf{F} \mathbf{Q}^{-1}$$

Lo que nos permite definir la función \mathbf{f} como transformada inversa de \mathbf{F} .

Nos concentraremos ahora en un tipo de transformada, la llamada transformada de Fourier. Para ello consideramos matrices \mathbf{W} , de orden $J \times J$, de la forma

$$W(p, q) = \frac{1}{J} \exp[-2\pi i(pq/J)] \quad 0 \leq p, q < J$$

y usamos

$$P(p, q) = \frac{1}{M} \exp[-2\pi i(pq/M)] \quad 0 \leq p, q < M \quad (4.17)$$

$$Q(p, q) = \frac{1}{N} \exp[-2\pi i(pq/N)] \quad 0 \leq p, q < N \quad (4.18)$$

tendremos entonces que

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

y la llamaremos *transformada de Fourier* de \mathbf{f} .

Es obvio que la siguiente pregunta es ¿Cuál es la transformada inversa?.

Si tenemos en cuenta que

$$\sum_{u=0}^{J-1} \exp[-2\pi i(\frac{lu}{J})] \exp[2\pi i(\frac{mu}{J})] = \begin{cases} 0 & \text{si } l \neq m \\ J & \text{si } l = m \end{cases} \quad (4.19)$$

entonces

$$P^{-1}(p, q) = \exp[2\pi i(pq/M)] \quad 0 \leq p, q < M \quad (4.20)$$

$$Q^{-1}(p, q) = \exp[-2\pi i(pq/N)] \quad 0 \leq p, q < N \quad (4.21)$$

Tenemos entonces que

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

que recibe el nombre de transformada de Fourier inversa.

Por último es muy importante tener en cuenta que aunque las transformadas de Fourier directa e inversa están definidas en la región $0 \leq p < M$ y $0 \leq q < N$, las fórmulas anteriores son aplicables a cualquier par de enteros, (k, l) , teniéndose que

$$F(u + pM, v + qN) = F(u, v) \quad \forall p, q \in \mathcal{Z}, 0 \leq u < M, 0 \leq v < N$$

y análogamente para las componentes de \mathbf{f} si las definimos a través de la transformada de Fourier inversa. En otras palabras, la transformada de Fourier hace hipótesis toroidales sobre la imagen.

Ejercicio. 3 *Comprobar experimentalmente, utilizando cantata, que si a una imagen le calculamos su transformada de Fourier y luego calculamos la inversa obtenemos la imagen original*

Ejercicio. 4 *Crear en cantata una imagen de tamaño 256×256 constante, calcular su espectro y visualizarlo utilizando cantata, examinar donde se encuentra localizado dicho espectro y donde está el origen de coordenadas en la representación.*

Ejercicio. 5 *Dada una imagen, calcular la magnitud del espectro. Analizar qué ocurre si antes realizamos un desplazamiento de la imagen al centro de la misma y a continuación una corrección toroidal, (no un completado con ceros). Visualizar todos los pasos intermedios.*

4.6.1. Convolución y Transformada de Fourier

Definimos la convolución bidimensional discreta como

$$g(\alpha, \beta) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h(m, n) f(\alpha - m, \beta - n) \quad 0 \leq \alpha < M, 0 \leq \beta < N \quad (4.22)$$

Se puede probar fácilmente que existe la siguiente relación entre las transformadas de Fourier de \mathbf{f} , \mathbf{g} y \mathbf{h} :

$$G(u, v) = MNF(u, v)H(u, v)$$

la demostración es como sigue:

$$\begin{aligned}
 g(\alpha, \beta) &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h(m, n) f(\alpha - m, \beta - n) \\
 &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left[\sum_{u=0}^{M-1} \sum_{v=0}^{N-1} H(u, v) \exp \left[2\pi i \left(\frac{um}{M} + \frac{vn}{N} \right) \right] \right] \\
 &\quad \times \sum_{w=0}^{M-1} \sum_{z=0}^{N-1} F(w, z) \exp \left[2\pi i \left(\frac{w(\alpha - m)}{M} + \frac{z(\beta - n)}{N} \right) \right] \\
 &= \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sum_{w=0}^{M-1} \sum_{z=0}^{N-1} \left[F(u, v) D(w, z) \exp \left[2\pi i \left(\frac{w\alpha}{M} + \frac{z\beta}{N} \right) \right] \right] \\
 &\quad \times \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \exp \left[2\pi i \left(\frac{(u-w)m}{M} + \frac{(v-z)n}{N} \right) \right] \\
 &= MN \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) H(u, v) \exp \left[2\pi i \left(\frac{u\alpha}{M} + \frac{v\beta}{N} \right) \right]
 \end{aligned}$$

lo que nos dice que la transformada de Fourier de una convolución es, salvo una constante, el producto de las transformadas de Fourier de las matrices que definen la convolución.

Ejemplo. 8 Supongamos que queremos calcular, para cada punto $0 \leq i < M$ y $0 \leq j < N$, el valor de

$$g(i, j) = \frac{1}{4} (f(i-1, j) + f(i+1, j) + f(i, j-1) + f(i, j+1)) \quad (4.23)$$

este cálculo podría hacerse punto a punto. Queremos, sin embargo, usar la transformada de Fourier. Obviamente, la fórmula (4.23) define una convolución con

$$h(0, 1) = h(0, -1) = h(1, 0) = h(-1, 0) = 1/4$$

usando (4.6) para \mathbf{h} tendremos que los valores de esta matriz sobre la región $0 \leq i < M$ y $0 \leq j < N$ son todos nulos salvo los siguientes

$$h(0, 1) = h(0, N-1) = h(1, 0) = h(M-1, 0) = 1/4$$

Esto nos dice que si calculamos las transformadas de Fourier de \mathbf{f} y \mathbf{h} tal y como están definidos en la región $0 \leq i < M$ y $0 \leq j < N$ tenemos, salvo una constante, la transformada de Fourier de \mathbf{g} , calculamos, entonces, la transformada inversa de la transformada de Fourier de \mathbf{g} y ya tenemos la convolución.

Puede probarse fácilmente que la transformada de Fourier de \mathbf{h} es

$$\frac{1}{MN} (\cos(2\pi u/M) + \cos(2\pi v/N))/2$$

Ejercicio. 6 Calcular la convolución de una imagen con una función de emborronamiento gaussiana utilizando transformada de Fourier en cantata.

Como puede comprobarse del ejercicio anterior, el cálculo de la transformada de Fourier de \mathbf{h} lleva asociado el escribir la matriz correspondiente de una forma bastante peculiar, es lógico preguntarse cómo se relaciona la transformada de Fourier de \mathbf{w} con la de \mathbf{h} cuando existe la siguiente relación entre ambos vectores

$$w(p, q) = h(p + l, q + m) \quad 0 \leq p < M \quad 0 \leq q < N$$

para algún (l, m) . Por ejemplo, para el caso que acabamos de describir podría ser más fácil calcular la transformada de \mathbf{h} definida mediante

$$h(0 + l, 1 + m) = h(0 + l, m - 1) = h(1 + l, m) = h(l - 1, m) = 1/4$$

con $l = M/2$ y $m = N/2$. Tenemos

$$\begin{aligned} W(u, v) &= \frac{1}{MN} \sum_{x,y} w(x, y) \exp \left[-2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \\ &= \frac{1}{MN} \sum_{x,y} h(x - l, y - m) \times \\ &\quad \exp \left[-2\pi i \left(\frac{u(x - l)}{M} + \frac{v(y - m)}{N} \right) \right] \exp \left[-2\pi i \left(\frac{ul}{M} + \frac{vm}{N} \right) \right] \\ &= H(u, v) \exp \left[-2\pi i \left(\frac{ul}{M} + \frac{vm}{N} \right) \right] \end{aligned}$$

En particular si $l = M/2$ y $m = N/2$ tendremos

$$W(u, v) = H(u, v)(-1)^{u+v}$$

Obviamente esta fórmula nos permite calcular la transformada de Fourier de \mathbf{h} a partir de la de \mathbf{w} .

4.6.2. Correlación

Antes de pasar a estudiar las relaciones entre convolución y matrices circulantes por bloques recordemos que se define la *correlación* entre \mathbf{f} y \mathbf{h} , mediante

$$g(\alpha, \beta) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h^*(m, n) f(\alpha + m, \beta + n) \quad 0 \leq \alpha < M \quad 0 \leq \beta < N \quad (4.24)$$

Puede probarse fácilmente que en este caso

$$G(u, v) = MNH^*(u, v)F(u, v) \quad (4.25)$$

donde * denota conjugado. Es obvio que si $h(m, n) = h(-m, -n) \forall m, n$, entonces la correlación coincide con la convolución y por tanto los coeficientes de Fourier de h han de ser reales ($H^*(u, v) = H(u, v), \forall u, v$).

4.7. Algunos ejemplos de Transformadas de Fourier

Una vez estudiada la relación entre la transformada de Fourier discreta y continua veamos algunos ejemplos de espectros de funciones así como alguna de sus propiedades. La figura 4.9 muestra lo que le ocurre a la transformada de Fourier cuando la imagen es rotada.

Ejercicio. 7 *calcular la transformada de Fourier de una imagen desplazada de otra, comparar los espectros.*

Podríamos decir que hasta aquí hemos estudiado en el capítulo el uso de transformada de Fourier como representación alternativa de una imagen o señal. En el resto del capítulo analizaremos su uso para la resolución de ecuaciones.

4.8. Uso de la transformada de Fourier para resolver sistemas de ecuaciones

4.8.1. Fourier, Convolución y Matrices Circulantes por Bloques

Vamos ahora a dar una interpretación y uso alternativo a la transformada de Fourier. Consideremos el siguiente conjunto de vectores

$$\{\mathbf{z}_{u,v} : 0 \leq u < M, 0 \leq v < N\}$$

estos vectores tienen tamaño $(M \times N) \times 1$, los estamos considerando como vectores y no como matrices, el orden puede ser el lexográfico por filas o por columnas. Sus componentes tienen el valor

$$z_{u,v}(m, n) = \exp\left\{2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right\} \quad 0 \leq m < M, 0 \leq n < N$$

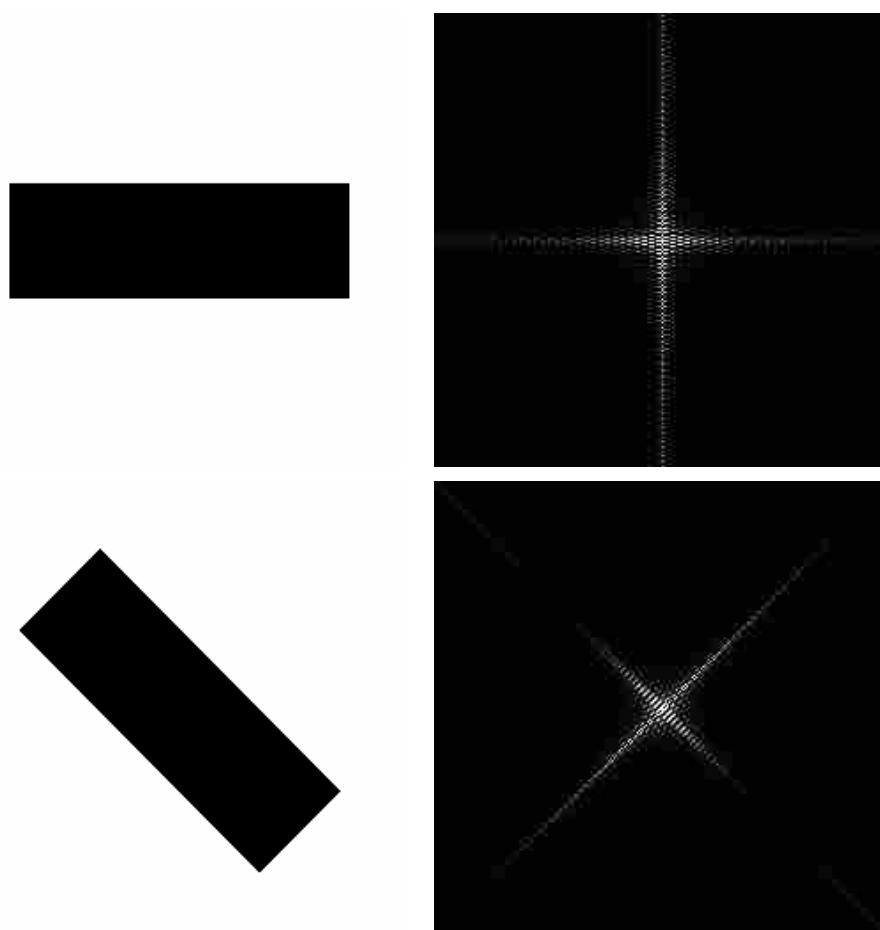


Figura 4.9: De izquierda a derecha y de arriba a abajo. Imagen y su espectro de potencias, la misma imagen rotada y su espectro de potencias

4.8. USO DE LA TRANSFORMADA DE FOURIER PARA RESOLVER SISTEMAS DE ECUACIONES

si ahora consideramos la transformada de Fourier desde este punto de vista, lo que nos está diciendo es que si consideramos como vector la imagen \mathbf{f} , con el mismo orden lexicográfico, e igualmente para su transformada de Fourier \mathbf{F} , podemos escribir

$$\mathbf{F} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \mathbf{z}_{x,y}^* \quad (4.26)$$

$$\mathbf{f} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \mathbf{z}_{u,v} \quad (4.27)$$

Consideremos la fórmula de la convolución (4.22) y escribamos \mathbf{g} y \mathbf{f} en orden lexicográfico, por ejemplo lexicográfico por filas, tendremos entonces que

$$\mathbf{g} = \mathcal{H}\mathbf{f}$$

donde \mathcal{H} es una matriz de orden $(M \times N) \times (M \times N)$, que es circular por bloques, es decir,

$$\mathcal{H} = \begin{pmatrix} \mathcal{H}_0 & \mathcal{H}_{-1} & \mathcal{H}_{-2} & \dots & \mathcal{H}_{-(M-1)} \\ \mathcal{H}_{-(M-1)} & \mathcal{H}_0 & \mathcal{H}_{-1} & \dots & \mathcal{H}_{-(M-2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{H}_{-1} & \mathcal{H}_{-2} & & & \mathcal{H}_0 \end{pmatrix}$$

con

$$\mathcal{H}_j = \begin{pmatrix} h(j, 0) & h(j, -1) & h(j, -2) & \dots & h(j, -(N-1)) \\ h(j, -(N-1)) & h(j, 0) & h(j, -1) & \dots & h(j, -(N-2)) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h(j, -1) & h(j, -2) & h(j, -3) & \dots & h(j, 0) \end{pmatrix}$$

siendo cada matriz \mathcal{H}_j de orden $N \times N$. Entender esta representación no es difícil. La primera fila de \mathcal{H} es la función de emborronamiento puesta según los coeficientes que van multiplicando a cada elemento de \mathbf{h} , para entender la segunda fila la dividimos en bloques de tamaño N , para cada bloque los coeficientes se obtienen de la línea anterior desplazando todos hacia adelante circularmente, con un ejemplo la contribución de $h(0, 0)$ a $h(0, 1)$ es la misma que la de $h(0, N-1)$ a $h(0, 0)$. Supongamos ahora que hemos completado las primeras N filas de D , en la fila $N+1$ los coeficientes que tenemos en el primer bloque (de los M) son los mismos que en el último bloque de la primera fila, (la fila primera es para la convolución de la segunda lo mismo que la última para la primera) y así sucesivamente.

Es importante tener en cuenta que $\mathcal{H}_j = \mathcal{H}_{j+M}$.

Veamos ahora cuál es el valor de $\mathcal{H}\mathbf{z}_{u,v}$. La componente l -ésima del vector resultante corresponde al producto escalar del vector $\mathbf{z}_{u,v}$ por la fila que comienza con la línea j -ésima dentro del bloque $-(M-k)$ -ésimo con $l = kM + j$.

El producto de la fila l -ésima de \mathcal{H} por el vector $\mathbf{z}_{u,v}$ vale pues

$$\begin{aligned} & \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} h(-(M-k) - x, -(N-j) - y) \exp \left[2\pi i \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \\ &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h(-(M-k) - x, -(N-j) - y) \\ & \exp \left[-2\pi i \left(\frac{u(-(M-k) - x)}{M} + \frac{(-(N-j) - y)v}{N} \right) \right] \\ & \times \exp \left[-2\pi i \left(\frac{u(M-k)}{M} + \frac{v(N-j)}{N} \right) \right] \\ &= MNH(u, v) \exp \left[2\pi i \left(\frac{uk}{M} + \frac{vj}{N} \right) \right] \end{aligned}$$

Esto nos dice que los autovalores de \mathcal{H} son los coeficientes de Fourier de la matriz de convolución multiplicados por el tamaño de la matriz.

Consideremos la correlación entre \mathbf{h} y \mathbf{f} como ha sido definida en (4.24). Si escribimos la correlación en forma matricial puede probarse que

$$\mathbf{g} = \mathcal{H}^{*t} \mathbf{f}$$

Los autovalores de \mathcal{H}^{*t} son ahora $MN H^*(u, v)$.

4.8.2. Propiedades del cálculo matricial usando coeficientes de Fourier

Supongamos que queremos calcular, considerando \mathbf{f} como un vector columna, $\mathbf{f}^{*t} \mathbf{f}$, tendremos

$$\mathbf{f}^{*t} \mathbf{f} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} F^*(u, v) F(r, s) MN \mathbf{z}_{u,v}^{*t} \mathbf{z}_{r,s} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) F(u, v) MN \quad (4.28)$$

Análogamente

$$\mathbf{f}^{*t} \mathcal{H}^{*t} \mathcal{H} \mathbf{f} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} F^*(u, v) F(r, s) MN \mathbf{z}_{u,v}^{*t} D^t D \mathbf{z}_{r,s} \quad (4.29)$$

$$= \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v)(MN)^2 H(u, v)H^*(u, v)F(u, v)MN \quad (4.30)$$

4.8.3. Resolución de Ecuaciones Lineales Mediante Transformada de Fourier

Una aplicación del uso de la transformada de Fourier es la resolución de ecuaciones lineales que llevan asociadas matrices circulantes.

Supongamos que queremos resolver la ecuación

$$\mathcal{A}\mathbf{x} = \mathbf{y} \quad (4.31)$$

donde \mathcal{A} es una matriz circulante por bloques, es decir, es del tipo de las matrices de convolución que hemos visto antes.

Tenemos, desarrollando por Fourier \mathbf{x} e \mathbf{y}

$$\mathcal{A}\left(\sum_u \sum_v X(u, v)\mathbf{z}_{u,v}\right) = \sum_u \sum_v Y(u, v)\mathbf{z}_{u,v} \quad (4.32)$$

de donde

$$\sum_u \sum_v (MN)D(u, v)X(u, v)\mathbf{z}_{u,v} = \sum_u \sum_v Y(u, v)\mathbf{z}_{u,v} \quad (4.33)$$

donde $(MN) \times (MN)$ es el tamaño de \mathcal{A} y $(MN)\{D(u, v)\}$ son los coeficientes de Fourier de la matriz de convolución asociada a \mathcal{A} .

Tendremos por tanto que,

$$(MN)D(u, v)X(u, v) = Y(u, v) \quad \forall u, v \quad (4.34)$$

o

$$X(u, v) = Y(u, v)/(MN)D(u, v) \quad \forall u, v \quad (4.35)$$

de donde calculando la transformada inversa de $\{X(u, v)\}$ habremos calculado \mathbf{x} .

4.8.4. Transformada de Fourier Rápida. Implementación y aplicaciones en procesamiento de imágenes

Existen muchas implementaciones de la transformada de Fourier, la transformada de Fourier rápida es aplicable cuando M y N son potencias de 2. Una implementación posible es la del libro [51] que discutiremos y aplicaremos en un ejemplo.

El formato de la subrutina es

```
void fourn( float data[], unsigned long nn[], int ndim, int isign)
```

Esta subrutina reemplaza **data**, que comienza en la posición 1, por su transformada de Fourier **ndim**-dimensional discreta multiplicada por el número de datos, si **isign** es -1. **nn**[1,...,**ndim**] es una matriz de enteros que contiene la longitud de cada dimensión (número de valores complejos), los cuales han de ser una potencia de dos. **data** es una matriz real de longitud dos veces el producto de estas longitudes en el que los datos se almacenan en matrices complejas multidimensionales; la parte real y la imaginaria de cada elemento en posiciones consecutivas y el índice más a la derecha es el que se mueve más rápidamente en la matriz. Si **isign** es 1, **data** es reemplazado por su transformada de Fourier inversa.

Veamos una aplicación de esta función. Supongamos que queremos calcular la solución **f** del sistema

$$(\lambda C + \mathcal{H}^t \mathcal{H})\mathbf{f} = \mathcal{H}^t \mathbf{g}$$

donde C es la matriz que expresa una convolución con coeficientes

$$c(0,0) = 1, \quad c(1,0) = c(-1,0) = c(0,-1) = c(0,1) = -0,25$$

Suponemos que el **h** correspondiente a \mathcal{H} está centrada alrededor del (0,0) y que g y h son vectores de tamaño MN almacenados por filas.

Para calcular la solución del sistema anterior procedemos como sigue.

```
/* suponemos que existen los include correspondientes
donde estan definidas, entre otras cosas, las funciones que usamos */

#define MAXSIZE 256

float g[MAXSIZE*MAXSIZE],
      prior [MAXSIZE*MAXSIZE],
      blur[MAXSIZE*MAXSIZE];
float complexblur[MAXSIZE*MAXSIZE*2],
      complexg[MAXSIZE*MAXSIZE*2];
```

4.8. USO DE LA TRANSFORMADA DE FOURIER PARA RESOLVER SISTEMAS DE ECUACIONES

```
main(argc,argv)
int argc;
char **argv;
{

    /* Habremos declarado las variables que necesitemos */

    /* En g tenemos la imagen observada y en blur el emborronamiento */

    /* suponemos que g y blur tienen el mismo tamaño */

    /* las dimensiones son nr2 (filas) y nc2 (columnas) ambos
       potencias de 2 */

    /* por tanto */

    dim[0]=nr2;
    dim[1] =nc2;

    size = nc2*nr2;

    real2complex(blur,complexblur,size);
    real2complex(g,complexg,size);

    foun(complexblur-1,dim-1,2,-1);
    foun(complexg-1,dim-1,2,-1);

    cal_coefprior (prior, nc2, nr2);

    complex_rest(complexg,complexim,complexblur,prior,lambda,nc2,nr2);

    foun(complexim-1,dim-1,2,1);

    complex2real(complexim, g, size);
```

```

    mult_image(g, g, 1.0/size, size);

}

void complex_rest(complexim,complexrest,complexblur,prior,lambda,nc2,nr2)

float *complexim, *complexrest, *complexblur, *prior, lambda;
int nc2,nr2;

/* Calculates in the frequency domain
 * blurt(complexim)/(lambda*prior + blurtblur)
 * We assume that blur is complex, so it comes in complex format,
 */
{

    int i, size = nr2*nc2;
    float *p1,*p2, *p3, *p4, aux, re1, re2, im1, im2, re, im;

    for(i=0,p1=complexim,p2=complexrest,p3=complexblur,p4=prior;
i<size; i++,p1+=2,p3+=2,p4++){
        re1 = *p1;
        re2 = *p3;
999    im1 = *(p1+1);
        im2 = *(p3+1);
        re = re1 * re2 + im1 * im2 ;
        im = - re1*im2 + re2*im1 ;
        aux = 1.0/((re2*re2 + im2*im2) + lambda>(*p4));
        *p2+= aux * re;
        *p2+= aux * im;
    }
}

```

4.8. USO DE LA TRANSFORMADA DE FOURIER PARA RESOLVER SISTEMAS DE ECUACIONES

```
void cal_coefprior(prior, nc2, nr2)
float *prior ;
int nc2,nr2;

/* Calculates the fourier transform of the prior model.
*/
{

float *p1;
int i,j, size=nr2*nc2;
double pi2;
float coc1, coc2;

pi2 = 8 * atan(1.0);
coc1 = pi2 / nr2;
coc2 = pi2 / nc2;

for(i = 0, p1 = prior; i < nr2; i++)
    for(j=0;j<nc2;j++)
        *p1++= 1.0-0.5*(cos(i*coc1)+cos(j*coc2));
}

void real2complex (realmatrix, complexmatrix, size)
float *realmatrix, *complexmatrix;
int size;
/* convert the real image realmatrix to complex format.
* the assumed size of realimage is size, so complexmatrix
* should be size*2.
*/
```



```
{

    int i;
    float *p1, *p2;

    for (i = 0, p1 = realmatrix, p2 = complexmatrix; i < size; i++){
*p2++ = *p1++;
*p2++ = 0.0;
    }
}
```

```
void complex2real (complexmatrix, realmatrix, size)
float *complexmatrix, *realmatrix;
int size;
/* copies the real part of the complex matrix complexmatrix to
 * realimage, size is the size of realimage.
 * complexmatrix is stored in the format, real part, i part, .....
 * See the book "recipes in C" for details.
 */
```

```
{

    int i;
    float *p1, *p2;

    for (i = 0, p1 = realmatrix, p2 = complexmatrix; i < size; i++){
*p1++ = *p2++;
p2++;
    }
}
```

```
void mult_image (destination, source, factor, size)
```

```
float *destination, *source, factor;
int size;
/* Multiplies source by factor and
 * copies the result in destination.
 * the assumed size of both images is
 * size.
 */
{

    int i;
    float *p1, *p2;

    for (i=0,p1=destination,p2=source; i<size; i++)
        *p1++ = *p2++ * factor;
}
```

4.9. Ejercicios

1. Escribir un programa en C que calcule el espectro de potencias de una imagen y lo almacene en otra imagen.
2. Escribir un programa en C que realice la operación de convolución usando transformada de Fourier. Tanto la imagen como el emborronamiento serán introducidos como imágenes.
3. Escribir un programa en C que sirva para comprobar experimentalmente la veracidad de las ecuaciones 4.28,4.30.

Capítulo 5

Restauración de Imágenes

En el campo del análisis de imágenes, éstas son procesadas para mostrar o extraer información útil para el problema que estemos resolviendo. Como ya sabemos, debido a las imperfecciones en su proceso de captación las imágenes observadas son versiones degradadas de la imagen original. Esta degradación, que fue examinada anteriormente, es debida por una parte al emborronamiento y por otra al proceso de ruido.

El campo de la restauración de imágenes ([26]) se dedica a la estimación de la imagen original, la imagen sin corrupción que ha dado lugar a la imagen degradada. Básicamente este proceso es el inverso del sistema de formación de la imagen. En este tema veremos los modelos más simples de restauración de imágenes y el uso de la transformada de Fourier para estimar la imagen original.

5.1. Restauración sin condiciones, Filtrado Inverso

Como dijimos al principio del tema el objetivo de la restauración de imágenes es estimar la imagen original \mathbf{f} , dada una imagen degradada \mathbf{g} y algún conocimiento sobre la matriz de emborronamiento \mathbf{D} y el ruido ϵ , supuesto el modelo

$$\mathbf{g} = \mathbf{D}\mathbf{f} + \epsilon \quad (5.1)$$

El objetivo, ver [27] es buscar un estimador de la imagen original que haga mínima una determinada función. La primera idea en que puede pensarse es definir el estimador de la imagen original, $\hat{\mathbf{f}}$, como la solución de

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{z}} \{J(\mathbf{z}) = \|\mathbf{g} - \mathbf{D}\mathbf{z}\|^2\} \quad (5.2)$$

La solución de la ecuación anterior es muy fácil, diferenciando J con respecto a \mathbf{f} tenemos que

$$-2\mathbf{D}^t(\mathbf{g} - \mathbf{D}\hat{\mathbf{f}}) = 0$$

de donde tenemos

$$\hat{\mathbf{f}} = (\mathbf{D}^t\mathbf{D})^{-1}\mathbf{D}^t\mathbf{g}$$

Si la matriz inversa de \mathbf{D} existe, tendremos que

$$\hat{\mathbf{f}} = \mathbf{D}^{-1}\mathbf{g}$$

Es importante recordar que sabemos resolver, usando transformada de Fourier, estas ecuaciones para algunos tipos de matriz de emborronamiento, en concreto las que vienen de una convolución que son circulantes por bloques.

5.1.1. Restauración con restricciones

Cuando la inversa no existe o bien debido al proceso de ruido la estimación es inestable y no realista, la solución del sistema anterior requiere que introduzcamos restricciones en la solución que buscamos, estas restricciones suelen tener la forma de cotas superiores sobre las derivadas primera o segunda.

Sea \mathbf{Q} un operador lineal sobre \mathbf{f} . Consideremos ahora el problema de minimizar $\|\mathbf{Q}\mathbf{f}\|^2$, sujeto a la restricción

$$\|\mathbf{g} - \mathbf{D}\mathbf{f}\|^2 \leq \lambda^2 \quad (5.3)$$

En este caso nuestro estimador verifica

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{z}} \{J(\mathbf{z}) = \|\mathbf{Q}\mathbf{z}\|^2 + \alpha \|\mathbf{g} - \mathbf{D}\mathbf{z}\|^2\} \quad (5.4)$$

donde α es una constante llamada el multiplicador de Lagrange.

Diferenciando la ecuación anterior con respecto a \mathbf{f} tenemos que nuestro estimador verificará:

$$\hat{\mathbf{f}} = (\mathbf{D}^t\mathbf{D} + \beta\mathbf{Q}^t\mathbf{Q})^{-1}\mathbf{D}^t\mathbf{g} \quad (5.5)$$

con $\beta = 1/\alpha$ que ha de ser estimada para verificar la condición impuesta en (5.3). La forma de la matriz \mathbf{Q} depende del problema, pero podría ser la clásica que en cada punto evalúa

$$\mathbf{f}(i, j) - (\mathbf{f}(i-1, j) + \mathbf{f}(i+1, j) + \mathbf{f}(i, j-1) + \mathbf{f}(i, j+1))/4$$

para la cual (5.5) puede resolverse por transformada de Fourier.

5.2. Filtro de Wiener

Consideremos el proceso de observación ya descrito

$$\mathbf{g} = \mathbf{D}\mathbf{f} + \epsilon$$

donde \mathbf{D} es una matriz circulante por bloques. El objetivo es encontrar $\hat{\mathbf{f}}$ definido por

$$\hat{\mathbf{f}} = \mathbf{A}\mathbf{g}$$

que minimice

$$E[\|\mathbf{f} - \hat{\mathbf{f}}\|^2] \quad (5.6)$$

En palabras, estamos buscando la mejor estimación lineal de la imagen original basada en la imagen observada.

Minimizar la ecuación (5.6) es equivalente a que se cumpla la siguiente condición de ortogonalidad

$$E[(\mathbf{f} - \hat{\mathbf{f}})\mathbf{g}^t] = 0 \quad (5.7)$$

lo que produce

$$E[\mathbf{f}\mathbf{g}^t] - \mathbf{A}E[\mathbf{g}\mathbf{g}^t] = 0 \quad (5.8)$$

ahora bien, ya que

$$\begin{aligned} E[\mathbf{f}\mathbf{g}^t] &= R_{\mathbf{f}}\mathbf{D}^t \\ E[\mathbf{g}\mathbf{g}^t] &= \mathbf{D}R_{\mathbf{f}}\mathbf{D}^t + R_{\epsilon} \end{aligned}$$

donde $R_{\mathbf{y}}$ denota la matriz de covarianzas del vector \mathbf{y} , tendremos que la matriz \mathbf{A} verificará:

$$\mathbf{A} = R_{\mathbf{f}}\mathbf{D}^t(\mathbf{D}R_{\mathbf{f}}\mathbf{D}^t + R_{\epsilon})^{-1} \quad (5.9)$$

Es muy interesante tener en cuenta la forma que tienen las matrices de covarianza para algunos problemas. Por ejemplo, consideremos una distribución para la imagen de la forma

$$p(\mathbf{f}) \propto \exp[-\alpha\mathbf{f}^t(I - C)\mathbf{f}]$$

siendo

$$\mathbf{f}^t(I - C)\mathbf{f} = \sum_i \sum_j f(i, j)(f(i, j) - 0,25[f(i - 1, j) + f(i + 1, j) + f(i, j - 1) + f(i, j + 1)])$$



Figura 5.1: Imagen observada de Saturno a la izquierda y restaurada a la derecha

observemos que $I - C$ es una matriz circulante por bloques que corresponde a una convolución de la forma

$$h(0,0) = 1, h(0,1) = h(0,-1) = h(1,0) = h(-1,0) = -0,25$$

además la matriz de covarianza de \mathbf{f} es $[\alpha(I - C)]^{-1}$ por otra parte, si consideramos ruido independiente tendremos

$$R_\epsilon = \beta I$$

Lo importante es que (5.9) puede resolverse usando transformada de Fourier dando la solución

$$\hat{F}(u, v) = \frac{1}{\lambda L(u, v) + D^*(u, v)D(u, v)} D^*(u, v) G(u, v)$$

donde $L(u, v)$ son los coeficientes de Fourier de la convolución asociada a \mathbf{h} multiplicados por el tamaño de la imagen y $D(u, v)$ son los coeficientes de Fourier de la matriz \mathbf{D} también multiplicados por el tamaño de la imagen y $\lambda = \alpha/\beta$.

Un ejemplo de restauración de imágenes se muestra en la figura 5.1

5.3. Ejercicios

1. Escribir un programa en C que calcule la solución de la ecuación 5.5. El formato del programa puede ser

```
restaura -i inputimage -b psf -l u -o outputimage
```

donde la imagen de entrada es `inputimage`, la de salida `outputimage`, el emborronamiento lo proporciona la imagen `psf` y $u = \beta/(\beta + 1)$ en la ecuación 5.5.

Capítulo 6

Mejora de Imágenes

6.1. Introducción

El principal objetivo de las técnicas de mejora de imágenes es procesar una imagen dada de forma que la imagen resultante sea más apropiada que la imagen original para unas aplicaciones específicas. Es importante destacar el término “específico” ya que establece que las técnicas que veremos en este capítulo estarán muy orientadas a aplicaciones concretas. Así por ejemplo, un método que es muy útil para mejorar imágenes de rayos X puede no ser la mejor aproximación para mejorar las imágenes de Marte transmitidas por un transbordador.

6.2. Mejora Mediante Procesamiento Puntual

Podría decirse que este tipo de transformaciones son operaciones de memoria cero (ver [21]), en el sentido de no tener en cuenta información local. Un nivel de gris $u \in [0, L]$ se transforma en otro nivel de gris v , usualmente en el mismo rango, mediante

$$v = T(u)$$

6.2.1. Ampliación de Contraste

Las imágenes con poco contraste aparecen a menudo debido a iluminación pobre o no-uniforme o a la no linealidad o pequeño rango dinámico de los sensores de las imágenes. Un ejemplo típico de transformación se muestra en la figura 6.2.1 que puede expresarse

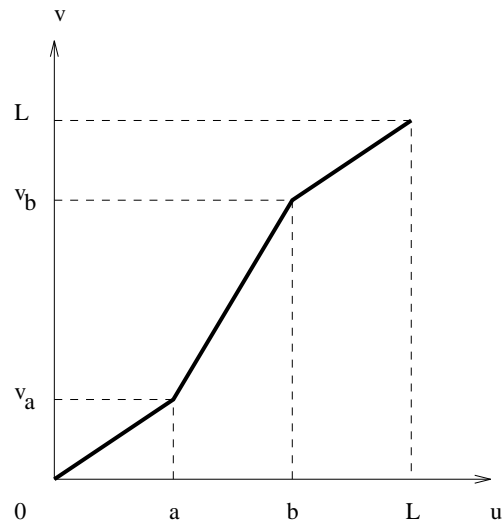


Figura 6.1: Ampliación del Contraste

mediante

$$v = \begin{cases} \alpha u & 0 \leq u < a \\ \beta(u - a) + v_a & a \leq u < b \\ \gamma(u - b) + v_b & b \leq u < L \end{cases}$$

las pendientes se toman mayores que uno en las regiones en las que se quiere ampliar el contraste.

Los parámetros a y b se pueden estimar examinando el histograma de la imagen. Por ejemplo, los intervalos de nivel de gris donde los píxeles ocurren con mayor frecuencia deben ampliarse para mejorar la visibilidad de la imagen.

6.2.2. Corte y Umbralización

Un caso particular del anterior corresponde a tomar $\alpha = \gamma = 0$ que recibe el nombre de corte. Este corte puede ser útil por ejemplo cuando sabemos que la señal está en el rango $[a, b]$.

La umbralización es un caso especial del modelo de corte en el que se obtiene una

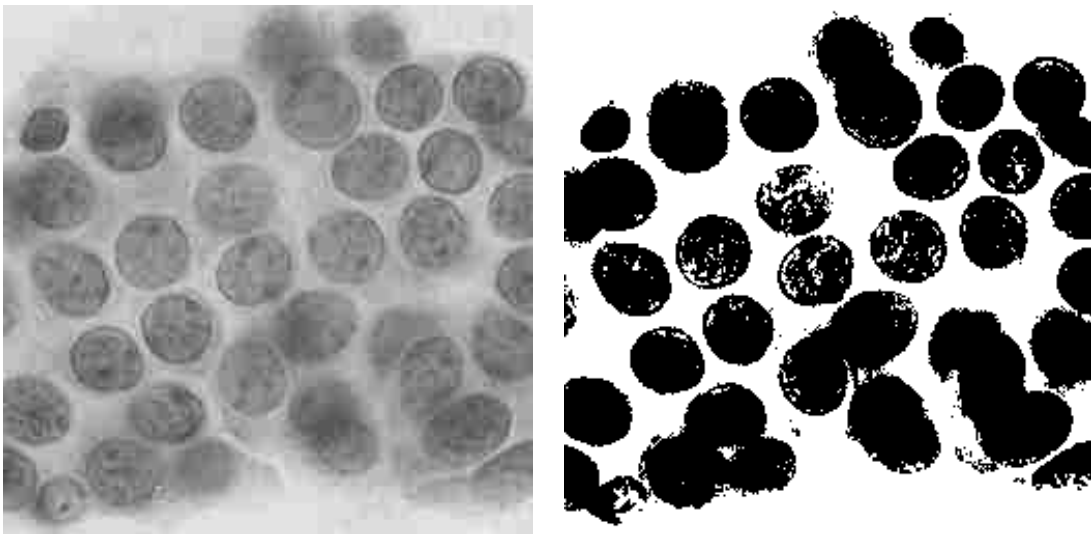


Figura 6.2: Imagen original y su umbralización por 150

imagen binaria. Este puede expresarse como

$$v = \begin{cases} 0 & 0 \leq u < a \\ L & a \leq u < b \\ 0 & b \leq u < L \end{cases}$$

Este modelo puede ser útil cuando sabemos que la imagen ha de ser binaria y por el proceso de digitalización no se ha obtenido así. También puede ser usado en un proceso de segmentación, ver figura 6.2

6.2.3. Negativo Digital

Una imagen negativa puede obtenerse cambiando la escala mediante la transformación

$$v = L - u$$

Este tipo de transformación es útil cuando en imágenes médicas y para la producción de negativos de imágenes.

6.2.4. Extracción de Bits

Supongamos que cada pixel en una imagen es cuantificada uniformemente en B bits. Deseamos extraer el n -ésimo bit más significativo y mostrarlo. Sea

$$u = k_1 2^{B-1} + k_2 2^{B-2} + \dots + k_n 2^{B-n} + \dots + k_{B-1} 2^1 + k_B$$

Queremos que el resultado sea

$$v = \begin{cases} L & \text{si } k_n = 1 \\ 0 & \text{en caso contrario} \end{cases}$$

Este tipo de transformación es útil para determinar el número de bits que son visualmente significativos, ya que el display de determinados planos puede probar que ellos no son significativos.

6.2.5. Compresión del Rango

A veces el rango dinámico de una imagen puede ser muy grande. Son ejemplos típicos las imágenes astronómicas o las de su función de emborronamiento. El rango dinámico puede comprimirse mediante la transformación.

$$v = c \log_{10}(1 + u)$$

donde c es una constante para escalado. Este tipo de transformaciones realiza las pequeñas magnitudes de los píxeles comparada con las grandes magnitudes.

6.2.6. Potenciación

Otro tipo de transformación que puede ser útil y que en algunos casos corresponde a la respuesta de determinados medios de captación de imágenes es la transformación de la forma

$$v = c'(u/c)^\lambda$$

c' es una constante de escalado mientras que c lleva el rango inicial al intervalo $[0, 1]$ y $\lambda \in [0, \infty]$. Si $\lambda < 1$ amplía los niveles de gris bajos mientras que si $\lambda > 1$ el efecto es el contrario.

6.2.7. Diferencia de Imágenes

La diferencia entre dos imágenes $f(x, y)$ y $g(x, y)$ viene expresada por

$$o(i, j) = f(i, j) - g(i, j) \quad \forall i, j$$

Esta técnica tiene numerosas aplicaciones en segmentación y realce.

6.2.8. Promedio de Imágenes

Supongamos que el valor del ruido ϵ en cada pixel es una variable aleatoria con media cero y desviación típica σ . Es decir, tenemos el modelo

$$g(i, j) = f(i, j) + \epsilon(i, j) \quad (6.1)$$

para todo i, j . El resultado de realizar el promedio sobre n imágenes distintas de la misma escena sería

$$\frac{f + \dots + f}{n} + \frac{\epsilon_1 + \dots + \epsilon_n}{n} \quad (6.2)$$

Lo que ocurre es que ahora la desviación típica del ruido es σ/\sqrt{n} . Por tanto si tenemos n imágenes de la misma escena, podemos obtener una nueva versión de la imagen original con menos ruido mediante

$$\hat{f}(i, j) = \frac{1}{n} \sum_{k=1}^n g_k(i, j) \quad (6.3)$$

6.3. Modificación del Histograma

Aunque esta técnica podría entenderse como una de procesamiento puntual, sus características especiales hacen aconsejable su separación y estudio independiente.

El histograma de una imagen digital con niveles de gris en el rango $[0, L - 1]$ es una función discreta con valores $p(r_k) = n_k/n$ donde r_k es el k -ésimo nivel de gris, n_k es el número de píxeles en la imagen con ese nivel de gris, n es el número total del píxeles en la imagen y $k = 0, 1, 2, \dots, L - 1$.

Podría decirse que $p(r_k)$ proporciona una estimación de la probabilidad de ocurrencia del nivel de gris r_k . Un gráfico de esta función proporciona una descripción muy simple de la apariencia de la imagen. La figura 6.3 muestra los histogramas de cuatro tipos básicos de imágenes. El histograma mostrado en la figura 6.3(a) muestra que los niveles de gris están concentrados en la zona oscura de la escala. Lo contrario ocurre en la figura 6.3(b).

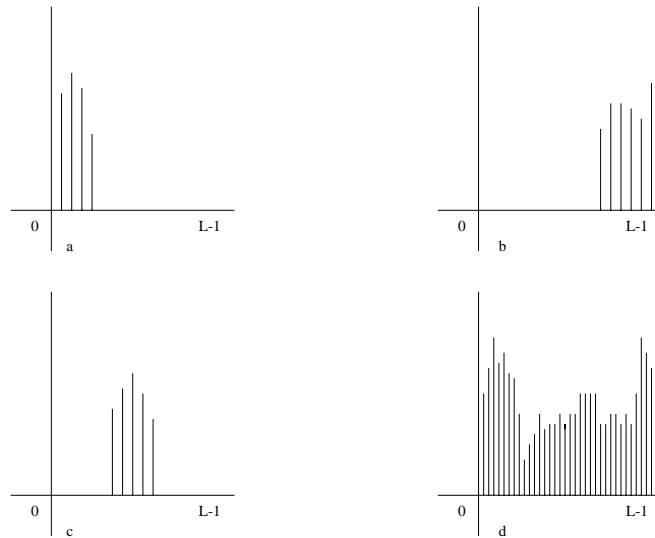


Figura 6.3: Ejemplos de histograma

Un histograma con poco rango dinámico, y por tanto de bajo contraste es mostrado en la figura 6.3(c). Finalmente, la figura 6.3(d) corresponde a alto contraste. Aunque las propiedades que acabamos de discutir son descripciones globales que no dicen nada concreto sobre el contenido de la imagen, la forma del histograma nos proporciona información sobre las posibilidades de mejora del contraste. Veamos ahora formas de manipular el histograma que son consistentes.

6.3.1. Uniformización del Histograma

Sean r los niveles de gris en la imagen que pretendemos mejorar. En el comienzo de la discusión supondremos que los valores de los píxeles son cantidades continuas que han sido normalizadas para que se extiendan en el rango $[0, 1]$, con $r = 0$ representando negro y $r = 1$ representando blanco. Posteriormente consideraremos la formulación discreta y permitiremos el rango $[0, L - 1]$.

Para cualquier r en el intervalo $[0, 1]$ nos centraremos en transformaciones de la forma

$$s = T(r) \tag{6.4}$$

que producen un nivel de gris s para cada nivel de gris r de la imagen original. Se supone que la transformación que se usa en la ecuación (6.4) cumple las condiciones siguientes

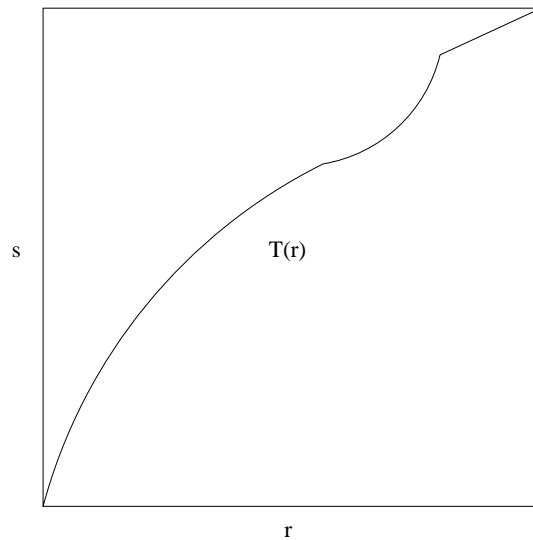


Figura 6.4: Una transformación de los niveles

1. $T(r)$ es una función univaluada monótonamente creciente en $0 \leq r \leq 1$ y
2. $0 \leq T(r) \leq 1$.

La condición primera conserva el orden de negro a blanco en la escala de gris, mientras que la segunda garantiza que la aplicación es consistente con el rango de niveles de gris permitido.

La figura 6.3.1 muestra un ejemplo de transformación que cumple dichas condiciones anteriores.

La transformación inversa de s a r se nota

$$r = T^{-1}(s) \qquad 0 \leq s \leq 1 \qquad (6.5)$$

Donde es obvio que esta transformación debe también cumplir las dos condiciones anteriores y en particular ser univaluada.

Los niveles de gris en una imagen pueden considerarse como cantidades aleatorias en el intervalo $[0, 1]$. Si son variables continuas, los niveles de gris original y transformado pueden caracterizarse por sus funciones de densidad de probabilidad $p_r(r)$ y $p_s(s)$, respectivamente.

Por resultados de la teoría de la probabilidad, si $p_r(r)$ y $T(r)$ son conocidos y $T^{-1}(s)$ satisface la primera condición anterior, la densidad de probabilidad del nivel de gris trans-

formado, supuesto que existe, es

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \quad (6.6)$$

Las técnicas que veremos ahora se basan en modificar la apariencia de una imagen controlando la función de densidad de probabilidad de sus niveles de gris mediante la transformación $T(r)$.

Consideremos la función de transformación

$$s = T(r) = \int_0^r p_r(w) dw \quad 0 \leq r \leq 1 \quad (6.7)$$

Esta transformación es simplemente la función de distribución de la variable r . es obvio que esta transformación satisface las condiciones que antes exigimos a toda transformación.

Además

$$\frac{ds}{dr} = p_r(r) \quad (6.8)$$

Tenemos, sustituyendo en la definición de la densidad de s , que

$$p_s(s) = 1 \quad 0 \leq s \leq 1 \quad (6.9)$$

que es la densidad uniforme en el intervalo de definición de la variable transformada s . Este resultado es independiente de la transformación inversa. Esto es importante puesto que no siempre es fácil obtener la transformación inversa.

El desarrollo anterior indica que usando una función de transformación igual a la función de distribución de r produce una imagen cuyos niveles de gris tienen una densidad uniforme. En términos de mejora de la imagen esto significa un crecimiento del rango dinámico de los píxeles, que puede mejorar considerablemente la apariencia de la imagen.

Antes de analizar como podemos aplicar esta técnica en el caso discreto, veamos un ejemplo de aplicación de las ecuaciones anteriores.

Supongamos que los niveles r tienen la función de densidad que se muestra en la figura 6.3.1(a). En este caso $p_r(r)$ es

$$p_r(r) = \begin{cases} -2r + 2 & 0 \leq r \leq 1 \\ 0 & \text{en el resto} \end{cases} \quad (6.10)$$

Consideremos la transformación de la función de distribución. En ese caso tendremos

$$s = \int_0^r (-2w + 2) dw = -r^2 + 2r \quad (6.11)$$

Aunque $T(r)$ es todo lo que necesitamos conocer para uniformizar el histograma, vamos a probar que esta nueva variable es de hecho uniforme. Tenemos

$$r = T^{-1}(s) = 1 \pm \sqrt{1-s} \quad (6.12)$$

Puesto que r está en el rango $[0, 1]$, la única solución es

$$r = T^{-1}(s) = 1 + \sqrt{1-s} \quad (6.13)$$

Entonces la función de densidad de s es

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} = \left[(-2r + 2) \frac{dr}{ds} \right]_{r=1-\sqrt{1-s}} \quad (6.14)$$

$$= \left[2\sqrt{1-s} \frac{d}{ds} (1 - \sqrt{1-s}) \right] = 1 \quad 0 \leq s \leq 1 \quad (6.15)$$

que es uniforme en el rango que queríamos. La figura 6.3.1(b) muestra la transformación $T(r)$ y la figura 6.3.1(c) la densidad resultante.

Para que los resultados anteriores sean útiles en el procesamiento digital de imágenes, los conceptos que hemos estudiado han de ser formulados en forma discreta. Para niveles de gris que toman valores discretos los convertimos en probabilidades mediante

$$p_r(r_k) = \frac{n_k}{M \times N} \quad 0 \leq r_k \leq 1 \text{ y } k = 0, 1, \dots, L-1 \quad (6.16)$$

donde L es el número de los niveles de gris, $p_r(r_k)$ es la probabilidad del k -ésimo nivel de gris y n_k es el número de veces que dicho nivel de gris aparece en la imagen y $M \times N$ es el número de píxeles en la imagen. Es obvio que nuestra transformación para uniformizar el histograma podría ser

$$S = T(r_k) = (L-1) \sum_{l=0}^k p_r(r_l) \quad (6.17)$$

La figura 6.3.1 muestra un ejemplo de igualación del histograma

6.3.2. Especificación del Histograma

Aunque el método que acabamos de discutir es bastante útil, no conduce a la aplicación de técnicas de mejora interactivas ya que este método es sólo capaz de generar un resultado, una aproximación a la distribución uniforme.

Es a veces deseable ser capaces de especificar interactivamente histogramas concretos que resalten determinados rangos de niveles de gris. Para ver como esto puede llevarse

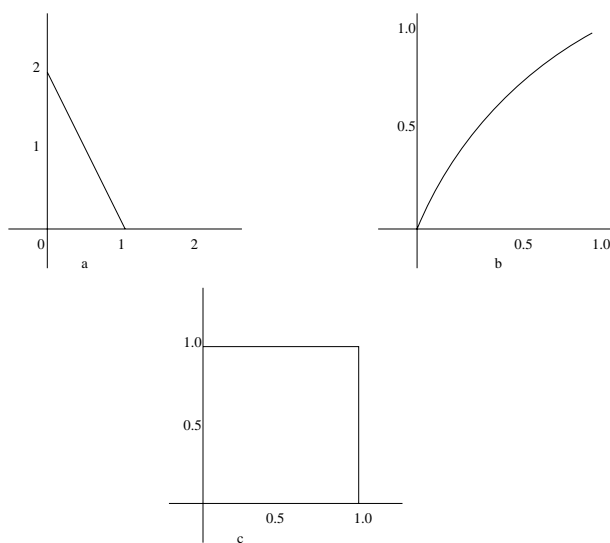


Figura 6.5: Ilustración de transformación uniforme

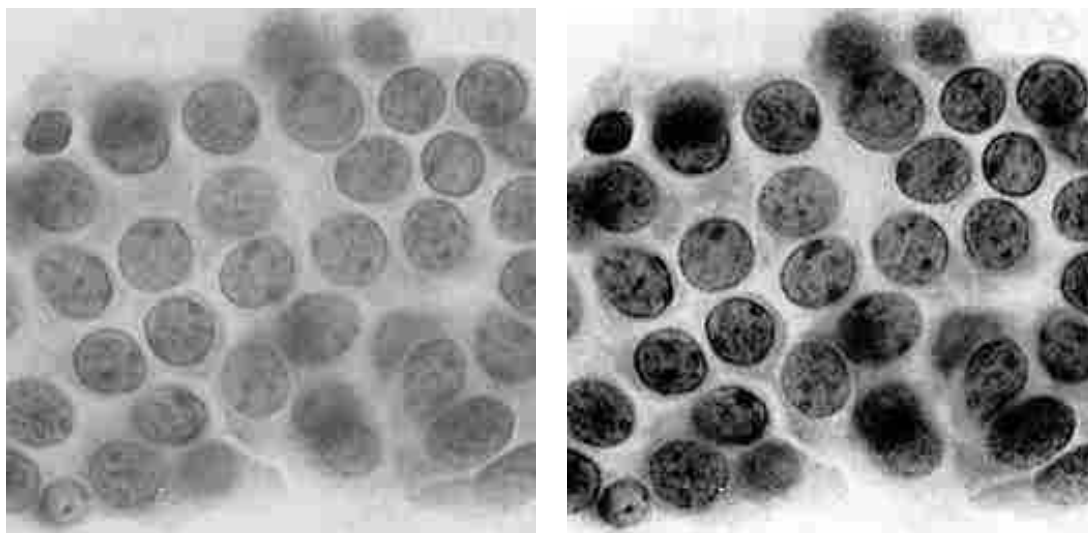


Figura 6.6: Ejemplo de igualación del histograma

a cabo, supongamos, en el caso continuo, que $p_r(r)$ y $p_z(z)$ son las densidades de probabilidad original y deseada respectivamente. Supongamos que a una imagen dada se le uniformiza el histograma utilizando

$$s = T(r) = \int_0^r p_r(w)dw$$

Obviamente si la imagen deseada estuviera disponible, sus niveles podrían ser uniformizados utilizando la función de transformación

$$v = G(z) = \int_0^z p_z(w)dw$$

El proceso inverso, $z = G^{-1}(v)$ produciría entonces los niveles que necesitamos. Observemos, por tanto, que lo que necesitamos es construir una distribución uniforme. Esto puede llevarse a cabo utilizando el s obtenido de la imagen original, los niveles resultantes $z = G^{-1}(s)$ tendrían entonces la función de densidad deseada. Supuesto que $G^{-1}(s)$ es univaluada el proceso puede resumirse como sigue

1. Uniformizar los niveles de la imagen original,
2. Especificar la función de densidad deseada y obtener la transformación $G(z)$ que la llevaría a una uniforme,
3. Aplicar la transformación inversa $z = G^{-1}(s)$ a los niveles que hemos obtenido en el paso primero.

6.3.3. Mejora Local

Las técnicas de procesamiento del histograma discutidas en las dos secciones anteriores son globales en el sentido de que los píxeles son modificados por una función de transformación basada en la distribución de los niveles de gris sobre la imagen completa. Es obvio que en algunos casos es necesaria una mejora local de los detalles de la imagen.

Las técnicas de procesamiento del histograma descritas antes son fácilmente adaptables a mejora local. El procedimiento es definir una ventana cuadrada o rectangular y centrarla sobre cada pixel aplicando las técnicas del histograma a esta región.

En lugar de usar el histograma, la mejora local puede basarse en las propiedades de los píxeles en un entorno del que estamos analizando. La intensidad media y la varianza son dos propiedades que se usan frecuentemente por su relevancia en la apariencia de la

imagen. Podríamos decir que la media es una medida del brillo medio mientras que la varianza mide el contraste.

Una transformación local típica basada en estos conceptos aplica la intensidad de una imagen $\mathbf{f}(x, y)$ en un nuevo nivel de gris $\mathbf{g}(x, y)$ mediante la realización de la siguiente transformación en cada punto (x, y) .

$$g(x, y) = A(x, y)[f(x, y) - m(x, y)] + m(x, y)$$

donde

$$A(x, y) = k \frac{M}{\sigma(x, y)} \quad 0 < k < 1$$

En esta formulación $m(x, y)$ y $\sigma(x, y)$ son los niveles de gris medio y la desviación típica calculada en un entorno del píxel (x, y) , M es la media global de $f(x, y)$ y k es una constante de contraste.

Los valores de las variables A , m y σ dependen del entorno del píxel (x, y) . La aplicación del factor de ganancia local $A(x, y)$ a la diferencia entre $f(x, y)$ y su media local amplifica las variaciones locales. En regiones con bajo contraste $A(x, y)$ es grande y recíprocamente. La media se vuelve a sumar para mantener el nivel medio de intensidad.

6.4. Filtrado Espacial

El uso de plantillas (máscaras) espaciales para el procesamiento de imágenes recibe usualmente el nombre de *filtrado espacial* (para diferenciarlo del *filtrado en el dominio de las frecuencias* que usa transformada de Fourier) y las máscaras reciben el nombre de filtros espaciales. En esta sección vamos a examinar filtros espaciales, lineales y no lineales para la mejora de la imagen.

Los filtros lineales están basados, como ya hemos visto, en transformaciones de la forma

$$g(i, j) = \sum_{m, n \in \mathcal{O}} h(i - m, j - n) f(m, n) \quad (6.18)$$

donde $f(i, j)$ es la imagen de entrada $g(i, j)$ es la imagen de salida y $h(i, j)$ es el llamado núcleo de convolución. Un retículo de vecinos \mathcal{O} rectangular es lo usual, con un número impar de filas y columnas asignándole valores también al píxel central.

El tipo de preprocesamiento que vamos a describir usa muy poco conocimiento a priori sobre el contenido de la imagen. Además el tamaño de \mathcal{O} es algo que puede variar sustancialmente en función del tamaño de los objetos en la imagen.

Los llamados filtros de *paso bajo* atenúan o eliminan componentes de alta frecuencia en el dominio de Fourier mientras que dejan las frecuencias bajas sin alterarlas. Como ya sabemos las frecuencias altas en el dominio de Fourier caracterizan las fronteras y detalles con alta variación en la imagen, así pues los filtros de paso bajo emborronan en algún sentido la imagen. El efecto contrario se obtiene con los filtros de *paso alto*. Por último tenemos los filtros que actúan sobre una determinada región de las frecuencias, este tipo de filtro no es de aplicación en la mejora de imagen.

Con independencia del tipo de filtro lineal usado, la aproximación básica es sumar los productos entre los coeficientes de la plantilla y las intensidades de los píxeles en la plantilla. Un tipo general de plantilla es el siguiente

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Si notamos los niveles de gris de los píxeles en la máscara como z_1, z_2, \dots, z_9 , la respuesta de la máscara lineal es

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9$$

Los filtros espaciales no lineales también operan sobre los vecinos. Sin embargo, en general sus operaciones están basadas directamente en los valores de los píxeles que se consideran y no usan coeficientes en la forma de los filtros lineales. Como veremos en la próxima subsección, la reducción puede ser alcanzada también con filtros no lineales que calculan la mediana de los niveles de gris. Otro ejemplo incluye el máximo de los 9 píxeles o el mínimo que pueden usarse para calcular los píxeles localmente más brillantes u oscuros respectivamente.

6.4.1. Filtros de Alisamiento

Los filtros de alisamiento se usan con el objetivo de emborronar o para reducir el ruido. El emborronamiento es usado como preprocesamiento para suprimir detalles pequeños antes de la extracción de objetos usualmente grandes, también puede usarse para unir segmentos. La reducción de ruido puede llevarse a cabo con emborronamiento de un filtro lineal y también por un filtro no lineal.

Filtros de Paso Bajo

Aunque el filtro espacial podría ser modelizado mediante una gaussiana o en general cualquier función de densidad o su aproximación continua, es usual restringirse a filtros de tamaño 3×3 o bien máscaras con forma de estrella. Es muy importante que la suma de los coeficientes sea la unidad.

Veamos con un ejemplo unidimensional que ocurre en el dominio de las frecuencias. Supongamos que $h(0) = h(1) = h(-1) = 1/3$, para una señal de tamaño M . Su transformada de Fourier es

$$F(u) = \frac{1}{M} \frac{1}{3} (1 + 2 \cos(2\pi u/M))$$

esta función casi no altera las frecuencias bajas (cerca del cero y al M) y disminuye mucho las cercanas a la mitad ($M/2$).

Un ejemplo típico es la siguiente máscara.

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

A veces se aumenta la fuerza del pixel central y la de aquellos que están más próximos teniendo

$$h = \frac{1}{10} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

$$h = \frac{1}{16} \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Promediando según un rango

El objetivo es promediar sobre aquellos píxeles que cumplen algún criterio tratando de evitar el emborronamiento.

Por ejemplo podemos usar

$$h(m, n) = \begin{cases} 1 & \text{si } g(i - m, j - n) \in [min, max] \\ 0 & \text{en caso contrario} \end{cases}$$

El rango puede ser incluso dependiente del nivel de imagen en el pixel central.

Alisamiento por medianas

En un conjunto de valores ordenados, la mediana es el valor central.

El filtrado por mediana es un método de alisamiento que reduce el emborronamiento de las fronteras. La idea es reemplazar el valor de la imagen en un punto por la mediana de

los niveles de brillo de sus vecinos. Es obvio que los puntos aislados de ruido son eliminados y que además no emborrona mucho las fronteras y que puede ser aplicado iterativamente.

Uno de sus mayores problemas es que si usamos un retículo rectangular dañaremos las líneas finas y los bordes de los objetos. Esto podría ser evitado si tomamos los vecinos sólo en las direcciones verticales y horizontales.

El alisamiento por mediana es un caso particular del filtrado por estadísticos ordenados, donde los valores de los vecinos se ordenan en una sucesión y un nuevo valor se le asigna al píxel central que es una combinación lineal de estos valores ordenados.

Filtros de medias no lineales

El filtro medio no lineal es una generalización de los promedios. Se define como

$$f(m, n) = u^{-1} \left(\frac{\sum_{(i,j) \in \mathcal{O}} a(i, j) u(g(m-i, n-j))}{\sum_{(i,j) \in \mathcal{O}} a(i, j)} \right)$$

Si los pesos son constantes el filtro recibe el nombre de homomórfico, algunos ejemplos son

1. Media aritmética, $u(g) = g$
2. Media armónica, $u(g) = 1/g$
3. Media Geométrica, $u(g) = \log g$

6.4.2. Filtros de Realce

El objetivo principal del realce es destacar los detalles finos de una imagen o mejorar detalles que han sido emborronados, bien por errores o por problemas en la adquisición de la imagen.

Filtrado Espacial de Paso Alto Básico

Es claro que este tipo de filtro puede obtenerse si tiene coeficientes positivos cerca de su centro y negativos en la periferia. Para una máscara 3×3 si vale positivo en el centro y negativo en el resto es suficiente.

Una implementación clásica de filtro de paso alto es la siguiente

-1/8	-1/8	-1/8
-1/8	1	-1/8
-1/8	-1/8	-1/8

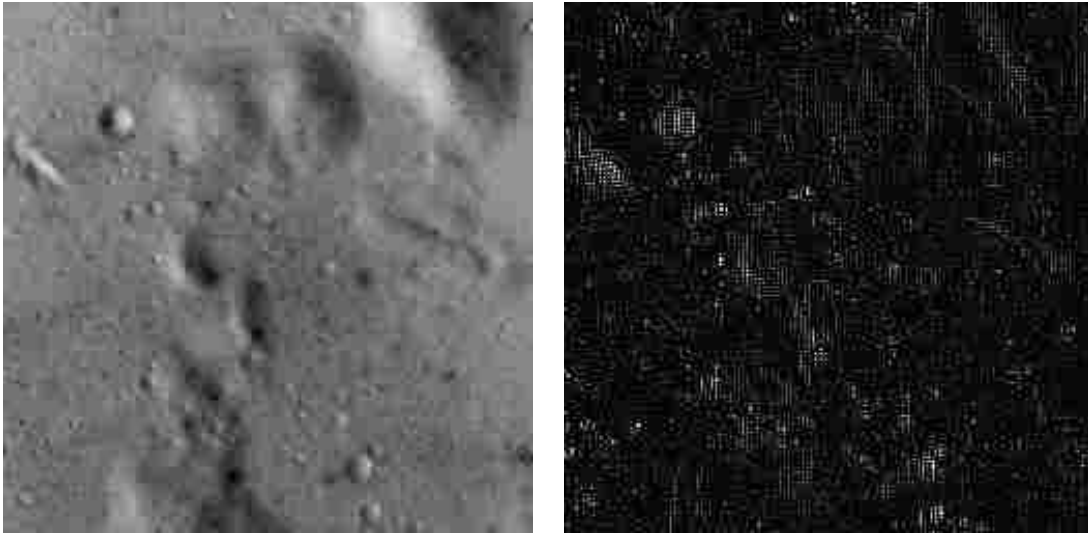


Figura 6.7: Imagen y su filtro de paso alto.

Es importante notar que la suma de los coeficientes es cero, así pues cuando la máscara se encuentra sobre una región constante o que varia muy suavemente el resultado es cero o muy pequeño.

Veamos su significado en el dominio de Fourier. Con un ejemplo unidimensional. Si $h(0) = 1$ y $h(-1) = h(1) = -1/2$ entonces la transformada de Fourier es

$$F(u) = \frac{1}{M}(1 - \cos(2\pi u/M))$$

que es cero, o próximo a él, en las frecuencias bajas y no modifica las altas.

El problema es que como este tipo de filtro produce una imagen de media cero hemos de buscar formas de llevarlo al rango de la imagen original (ver figura 6.7).

Filtros de Énfasis en las Frecuencias Altas

Una imagen filtrada de paso alto puede ser calculada como la diferencia entre la imagen original y una versión de paso bajo de la imagen, es decir

$$\text{Paso Alto} = \text{Original} - \text{Paso Bajo}$$

Si multiplicamos la imagen por un factor de amplificación, que notaremos A , produce lo que recibe el nombre de filtro de énfasis de las altas frecuencias (FEAF).

$$\begin{aligned} \text{FEAF} &= A \text{ Original} - \text{Paso Bajo} \\ &= (A-1)(\text{Original}) + \text{Paso Alto} \end{aligned}$$

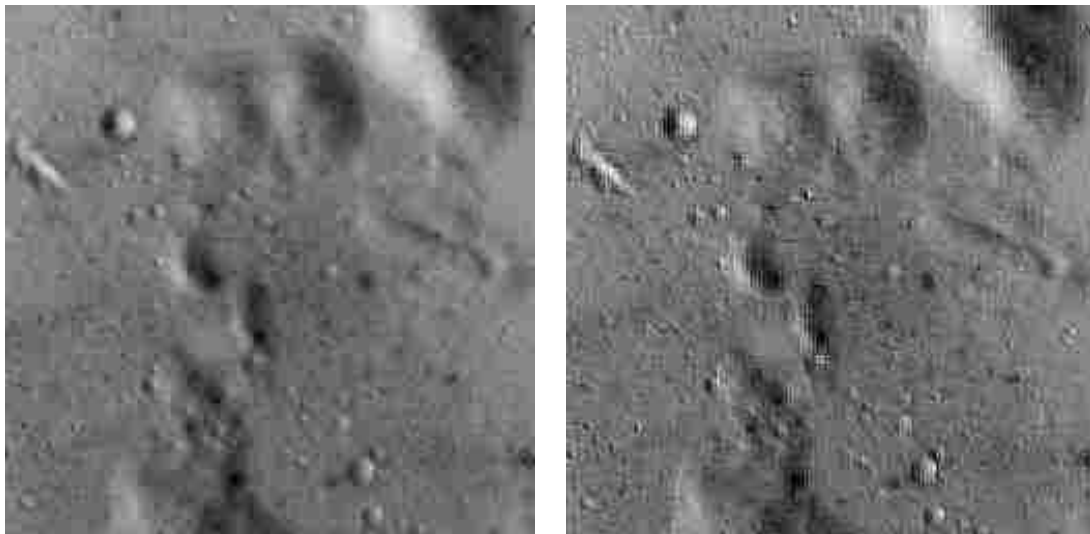


Figura 6.8: Imagen y su realce al

Cuando $A = 1$ produce el filtro clásico de paso alto. Cuando $A > 1$ parte de la imagen original se añade al filtro y por tanto se recuperan algunas de las bajas frecuencias. El resultado es que este tipo de filtro produce imágenes que se parecen más a la original con un relativo grado de mejora de bordes que depende del valor de A .

Este tipo de técnica se usa mucho en impresión y publicación de documentos.

En la misma línea pueden obtenerse resultados similares con filtros de la forma

$-w$	$-w$	$-w$
$-w$	$1 + 8w$	$-w$
$-w$	$-w$	$-w$

con $w > 0$, un ejemplo se muestra en la figura ??

6.5. Mejora en el Dominio de las Frecuencias

El fundamento de las técnicas basadas en el dominio de las frecuencias es el teorema de convolución que conocemos de capítulos anteriores. Este teorema dice que si $g(x, y)$ es la imagen formada por la convolución de la imagen $f(x, y)$ y un operador lineal invariante $h(x, y)$, es decir,

$$g(x, y) = (h * f)(x, y)$$

entonces por el teorema de convolución, se verifica la siguiente relación en el dominio de las frecuencias

$$G(u, v) = H(u, v)F(u, v)$$

donde G , H y F son las transformadas de Fourier de g , h y f respectivamente.

Muchos problemas de mejora de imágenes pueden ser formulados como sigue, dado $f(x, y)$, una vez calculado $F(u, v)$ el objetivo es seleccionar $H(u, v)$ de forma que la imagen deseada

$$g(x, y) = \mathcal{F}^{-1}[H(u, v)F(u, v)]$$

muestra algunos rasgos de interés de $f(x, y)$. Por ejemplo, bordes en la imagen $f(x, y)$ pueden ser acentuados usando una función que enfatice las frecuencias altas de $F(u, v)$.

$h(x, y)$ caracteriza un sistema cuyo objetivo es producir una imagen de salida $g(x, y)$ a partir de una imagen de entrada $f(x, y)$. El sistema realiza la convolución de $h(x, y)$ con la imagen de entrada y produce el resultado. El teorema de convolución permite dar una nueva interpretación al proceso, en el sentido de que se puede alcanzar el mismo resultado si multiplicamos $F(u, v)$ por $H(u, v)$ para producir $G(u, v)$. Calculando la transformada de Fourier inversa de la salida produce la imagen deseada.

Supongamos por un momento que $h(x, y)$ es desconocido y que aplicamos una función de impulso unitario (es decir, un punto de luz) al sistema. La transformada de Fourier del impulso unidad es simplemente uno, y por tanto tenemos

$$G(u, v) = H(u, v)$$

por tanto la transformada inversa de $G(u, v)$ es $h(x, y)$. Este resultado es bien conocido en la teoría de los sistemas lineales: un sistema lineal invariante por traslaciones se especifica completamente por su respuesta a un impulso.

Antes de pasar a estudiar la mejora de imagen desde el punto de vista del dominio de las frecuencias es importante notar que la mejora de imágenes no coincide con el problema de la restauración de las mismas, aunque su formulación podría resultar muy similar.

Tal y como acabamos de discutir, la mejora en el dominio de las frecuencias es, en principio, directa. Simplemente calculamos la transformada de Fourier de la imagen que queremos mejorar, multiplicamos el resultado por una función de transferencia, un filtro, y calculamos la transformada de Fourier inversa para producir la imagen mejorada.

Las ideas de emborronar mediante la reducción del contenido de las frecuencias altas o de realzar aumentando el contenido (la magnitud) de las frecuencias altas en relación

a las bajas vienen obviamente del dominio de Fourier. De hecho la idea de filtrado lineal se considera más atrayente e intuitiva en el dominio de las frecuencias. Sin embargo, en la práctica, las máscaras espaciales pequeñas se usan más que la transformada de Fourier por su simplicidad de implementación y velocidad en las operaciones. Sin embargo, la comprensión de los conceptos en el dominio de las frecuencias es esencial para la solución de problemas no fácilmente comprensibles en el dominio espacial.

6.5.1. Filtrado de Paso Bajo

Como ya indicamos antes, las fronteras y otras transiciones rápidas como el ruido en los niveles de gris de una imagen contribuyen significativamente al contenido de las frecuencias altas en el dominio de Fourier. Por tanto, el emborronamiento (alisamiento) se alcanza en el dominio de Fourier atenuando un rango especificado de componentes de altas frecuencias en la transformada de una imagen dada.

Como ya sabemos

$$G(u, v) = H(u, v) * F(u, v)$$

donde $F(u, v)$ es la transformada de Fourier de la imagen a alisar. El problema es seleccionar una función de transferencia $H(u, v)$ que produzca $G(u, v)$ atenuando las componentes de alta frecuencia de $F(u, v)$. La transformada inversa producirá entonces la imagen que queremos. En lo que sigue consideramos las funciones de transferencia que afectan la parte real e imaginaria de $F(u, v)$ de la misma forma.

Filtros Ideales

Un filtro ideal bidimensional es uno cuya función de transferencia cumple la relación

$$H(u, v) = \begin{cases} 1 & \text{si } D(u, v) \leq D_0 \\ 0 & \text{en caso contrario} \end{cases} \quad (6.19)$$

donde D_0 es una cantidad no negativa y $D(u, v)$ es la distancia del punto (u, v) al origen en el plano de las frecuencias, es decir,

$$D(u, v) = (u^2 + v^2)^{1/2}$$

El nombre ideal proviene de que todas las frecuencias dentro de un círculo de radio D_0 se pasan sin atenuación, mientras que todas las frecuencias fuera de ese círculo son atenuadas completamente.

Los filtros de paso bajo que consideramos en este capítulo son radialmente simétricos alrededor del origen, todas las frecuencias fuera de ese círculo son atenuadas.

Para un filtro ideal de paso bajo radialmente simétrico, el punto de transición entre $H(u, v) = 1$ y $H(u, v) = 0$ recibe usualmente el nombre de frecuencia de corte. La frecuencia de corte de un filtro ideal de paso bajo no puede realizarse con componentes electrónicas, aunque sí se puede simular con el ordenador.

El funcionamiento de todos los filtros de paso bajo que se introducen a través del dominio de Fourier se puede comparar examinando las frecuencias de corte. Una forma de establecer un conjunto de frecuencias de corte standard es calcular círculos que incluyen varias cantidades de la potencia de la señal P_T . Esta cantidad se obtiene como sigue

$$P_T = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} P(u, v)$$

donde $P(u, v)$ denota el espectro de potencias. Si la transformación ha sido centrada, un círculo de radio r con el origen en el centro de las frecuencias contiene β por ciento de la potencia

$$\beta = 100 \left[\sum_u \sum_v P(u, v) / P_T \right]$$

donde la suma se toma en los valores que están dentro del círculo o su frontera.

El emborronamiento y lo que se ha dado en llamar *ringing* de este tipo de filtros puede explicarse fácilmente en base al teorema de convolución. Puesto que la transformada de Fourier de la imagen original y la borrosa se relacionan en el dominio de Fourier mediante

$$G(u, v) = H(u, v)F(u, v)$$

el uso de este teorema conduce a la siguiente expresión en el dominio espacial

$$g(x, y) = (h * f)(x, y)$$

donde $h(x, y)$ es la transformada de Fourier inversa del filtro de transferencia $H(u, v)$.

El paso para entender el proceso de emborronamiento como un proceso de convolución en el dominio espacial está en la forma de $h(x, y)$. La discusión en el libro ([12]) es muy interesante.

Filtro de Butterworth

La función de transferencia del filtro de Butterworth de orden n y frecuencia de corte a distancia D_0 tiene la forma

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$

donde $D(u, v)$ es la distancia euclídea al centro de la imagen.

Al contrario que el filtro de paso bajo discutido antes el filtro de Butterworth no tiene una discontinuidad muy pronunciada que establece un corte claro entre las frecuencias que pasan y las que no pasan. La figura 6.5.1 muestra un ejemplo de la aplicación de este filtro.

6.5.2. Filtrado de Paso Alto

Como ya sabemos, una imagen puede ser emborronada atenuando las componentes de frecuencia alta en su transformada de Fourier. Esto es debido, como sabemos, a que los bordes y otros cambios abruptos en los niveles de gris se asocian con frecuencias altas. El realce de imágenes puede alcanzarse mediante el filtrado de paso alto que atenúa las frecuencias bajas sin distorsionar la información en las frecuencias altas en el dominio de Fourier.

Filtro Ideal

Un filtro bidimensional ideal de paso alto es uno cuya función de transferencia cumple la relación

$$H(u, v) = \begin{cases} 0 & \text{si } D(u, v) \leq D_0 \\ 1 & \text{en caso contrario} \end{cases}$$

donde D_0 es la distancia de corte medida desde el origen del plano de frecuencias y $D(u, v)$ ya ha sido definida.

Filtro de Butterworth

La función de transferencia de un filtro de Butterworth de paso alto viene definida por

$$H(u, v) = 1 - \frac{1}{1 + [D(u, v)/D_0]^{2n}} = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

donde las definiciones de los elementos que aparecen en él son las mismas que para el filtro de Butterworth de paso bajo.

6.6. Ejercicios

1. Implementar las técnicas de procesamiento puntual descritas en la sección 6.2.
2. Implementar la igualación del histograma.
3. Implementar un filtrado espacial para cualquier máscara de tamaño hasta 5×5 .

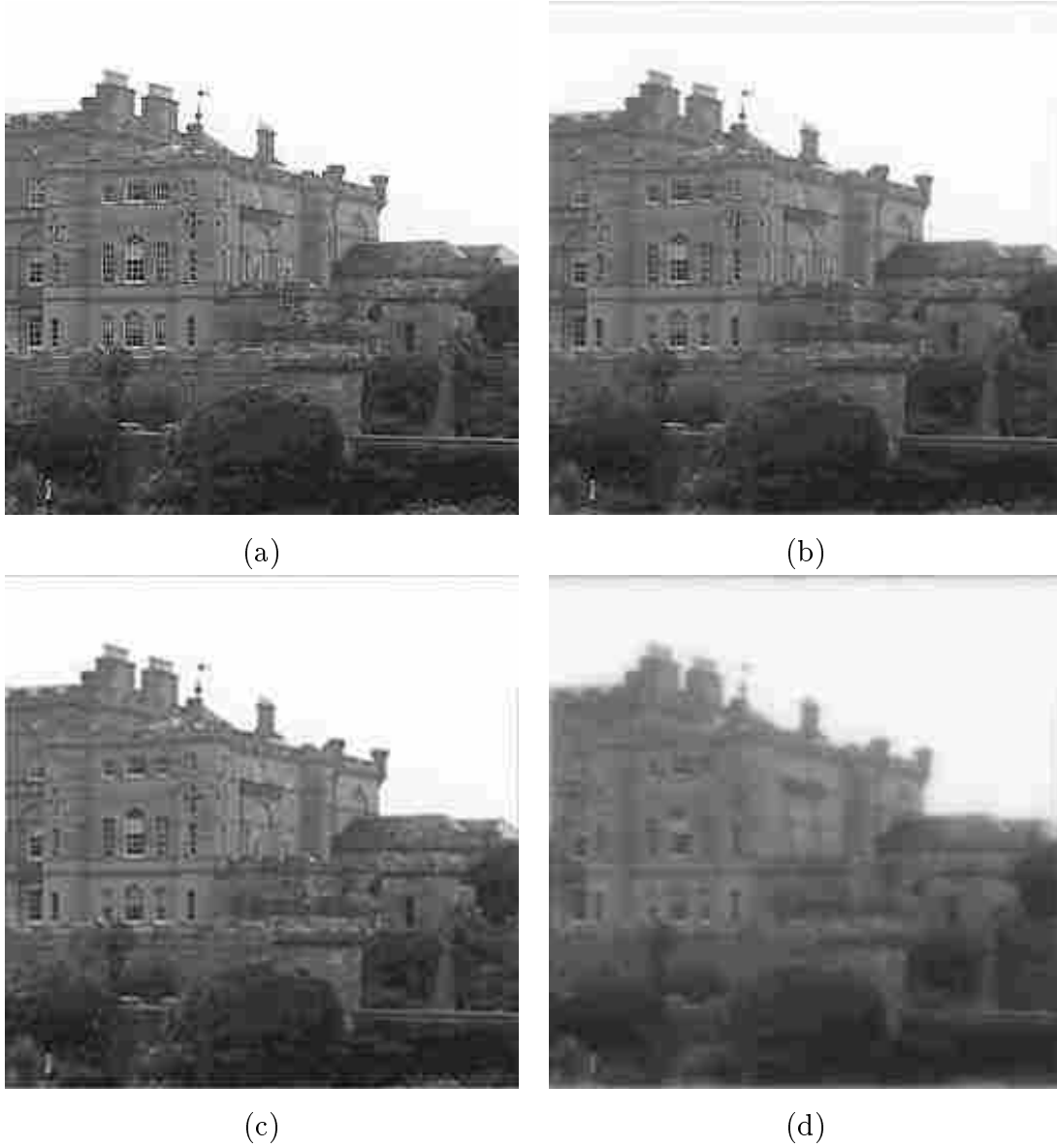


Figura 6.9: (a) Imagen original, (b) Butterworth de orden 1 y corte 0,5. (c) Butterworth de orden 6 y corte 0,5.(d) Butterworth de orden 1 y corte 0,1.

4. Implementar los filtros de Butterworth e ideales en el dominio de las frecuencias.

Capítulo 7

Segmentación Basada en Fronteras

Los primeros pasos de la visión tratan, como ya sabemos, de identificar rasgos en la imagen que son relevantes para estimar la estructura y propiedades de los objetos en la escena. Los bordes corresponden a cambios locales significativos en una imagen y son, probablemente, los rasgos más importantes para el análisis de la misma. Estos se encuentran en la frontera entre dos regiones diferentes, siendo su detección, por tanto, un paso importante para la recuperación de información. En este capítulo comenzaremos analizando la detección y localización de bordes, para en la segunda parte del mismo estudiar métodos simples para combinar dichos bordes en fronteras.

Simplificando la definición podríamos decir que un borde es una discontinuidad de algún tipo en la función de intensidad de la imagen. Pero, existen muchos tipos de bordes diferentes, algunos son debidos a las sombras que se producen sobre los objetos, otros a la variación de la reflectancia de los objetos o incluso por la textura de los objetos. Como vemos en la figura 7.1 los bordes etiquetados d son debidos a discontinuidades entre los objetos y la cámara, junto con discontinuidades en la normal a la superficie del objeto. Los etiquetados dc son debidos a discontinuidades de la distancia a la cámara pero no en la normal. Los etiquetados n son debidos a la discontinuidad en la normal, mientras que los etiquetados r se deben a cambios en la reflectancia de los objetos, pero sin cambios en las propiedades geométricas. Por último los etiquetados s se deben a la sombra que el cilindro produce sobre la mesa. Todos estos tipos distintos de procesos físicos pueden producir bordes. Si queremos relacionarlos con objetos en la escena tendremos que ser capaces de identificarlos con sus fuentes.

Teniendo en cuenta todo lo anterior podemos definir un borde como un cambio local

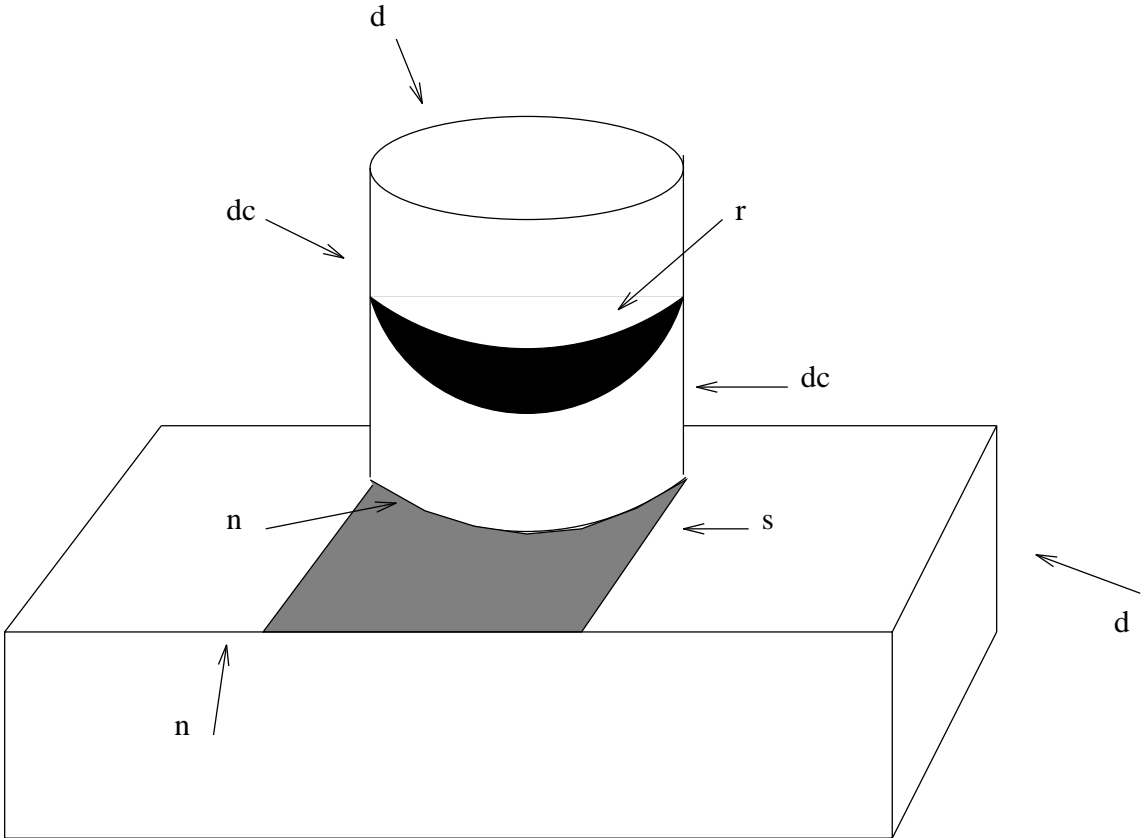


Figura 7.1: Tipos de Discontinuidades en una escena. Ver texto

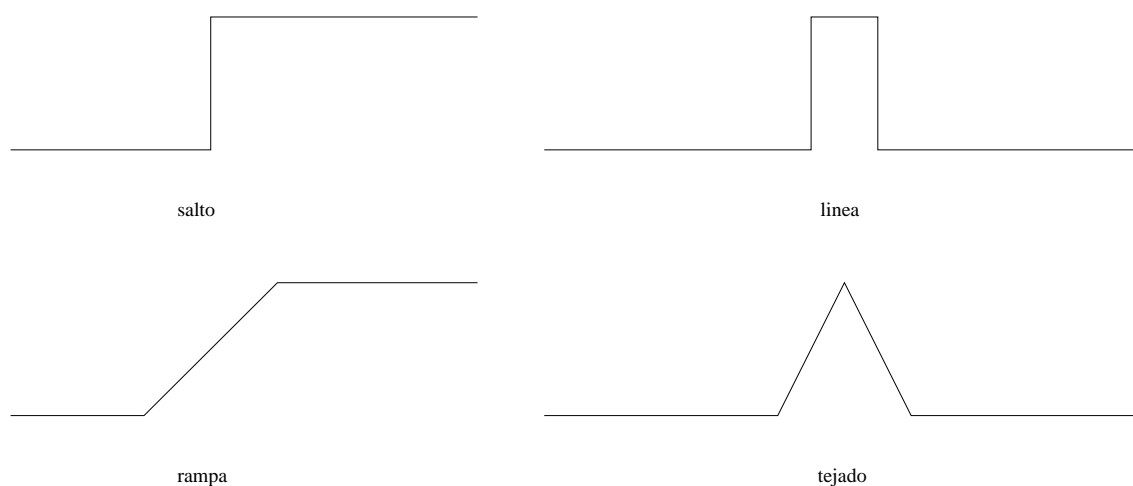


Figura 7.2: Perfiles de bordes. Modelos idealizados, fila superior y modelos observados, fila inferior.

significativo en la intensidad de la imagen que usualmente está asociado con una discontinuidad bien de la intensidad o su derivada. Las discontinuidades pueden ser de dos tipos: (1) de *salto* donde la intensidad cambia abruptamente de un valor en un lado de la discontinuidad a otro valor diferente en el lado opuesto, (2) discontinuidades de *linea*, donde las intensidades cambian abruptamente volviendo en poco espacio a su valor inicial. Es, sin embargo, claro que los saltos y pasos son raros en las imágenes (debido en parte al proceso de digitalización) y lo que se observa son rampas y tejados como se ve en la figura 7.2.

El tratamiento que haremos en este capítulo se centra principalmente en los bordes de salto aunque puede adaptarse fácilmente a otro tipo de cambios de intensidad en la imagen.

Antes de discutir algunos aspectos fundamentales sobre los detectores de bordes, veremos las siguientes definiciones ([43]).

Definición. 1 *Un punto de borde es un punto geométrico con coordenadas $[i, j]$ en la posición de un cambio de intensidad local significativo en la imagen.*

Definición. 2 *Un fragmento de borde es la i y j coordenada de un borde y la orientación θ que puede ser el ángulo del gradiente.*

Definición. 3 *Un detector de bordes es un algoritmo que produce un conjunto de bordes a partir de la imagen.*

Definición. 4 *Un contorno es una lista de bordes, o la curva matemática que modela la lista de bordes.*

Definición. 5 *La unión de bordes es el proceso de formar una lista ordenada de bordes a partir de una no ordenada.*

Definición. 6 *El seguimiento de bordes es el proceso de búsqueda en la imagen (filtrada) para determinar los contornos.*

Las coordenadas de un punto borde pueden ser los índices de la fila y la columna donde el borde fue detectado, o bien las coordenadas a otra resolución. Las coordenadas pueden venir dadas en el sistema de coordenadas de la imagen, pero más probablemente vendrán dadas en el sistema de coordenadas producido por el filtro de detección. El término borde se usa tanto para puntos como para fragmentos.

El conjunto de bordes que produce un detector de fronteras puede dividirse en dos clases: los correctos y los falsos. Un tercer conjunto corresponde a los bordes que deberían haber sido detectados. Este tipo de bordes reciben el nombre de falsos negativos mientras que los falsos corresponden a falsos positivos.

La diferencia entre la unión de fronteras y el seguimiento de fronteras es que la unión tiene como entrada un conjunto desordenado de bordes producido por un detector de bordes y el resultado es una lista ordenada. El seguimiento de bordes toma como entrada una imagen y produce una lista ordenada de bordes, este proceso puede utilizar información global.

7.1. Métodos de Gradiente

La detección de bordes es esencialmente una operación de detección de cambios de intensidad locales significativos. En una dimensión estos cambios, los de salto, están asociados con un pico local de la primera derivada. Veamos cuál es su equivalente en el caso bidimensional.

7.1.1. Operadores Isotrópicos

Un operador isotrópico tiene como objetivo determinar la fuerza de un borde con independencia de su dirección en la imagen. Es obvio que esta propiedad es deseable entre los detectores de bordes ya que no sería bueno que un detector sólo fuera sensible a bordes en determinadas direcciones. Este tipo de detectores es aplicado a cada punto de la imagen para calcular la fuerza del borde independientemente de su orientación. Veamos qué condiciones debe cumplir ese operador.

Consideremos la imagen de intensidades $f(x, y)$ con coordenadas x e y . Supongamos que la rotamos. Consideremos un punto (x, y) y su rotación por θ que nos produce (x', y') según la fórmula

$$\begin{aligned}x' &= x\cos\theta + y\sen\theta \\y' &= -x\sen\theta + y\cos\theta\end{aligned}$$

las derivadas de los dos sistemas se relacionan mediante

$$\frac{\partial f}{\partial x'} = \frac{\partial f}{\partial x}\cos\theta + \frac{\partial f}{\partial y}\sen\theta$$

y

$$\frac{\partial f}{\partial y'} = -\frac{\partial f}{\partial x}\sen\theta + \frac{\partial f}{\partial y}\cos\theta$$

Si uno está interesado en operadores isotrópicos, entonces un análisis matemático simple prueba que o bien derivadas de orden par o de orden impar elevadas a potencias pares pueden dar lugar a operadores isotrópicos. Puede probarse que un operador isotrópico es la suma de los cuadrados de las derivadas parciales en las direcciones x e y ya que

$$\left(\frac{\partial f}{\partial x'}\right)^2 + \left(\frac{\partial f}{\partial y'}\right)^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2$$

Para encontrar la dirección en la que se produce el cambio, la dirección del borde, consideremos la ecuación para la derivada parcial en la dirección x' . Podemos entonces obtener el θ para el que esta derivada parcial se maximiza. Para ello tendremos que resolver

$$d\left(\frac{\partial f}{\partial x'}\right)/d\theta = 0$$

de donde

$$\theta = \arctg\left(\frac{\partial f}{\partial y}/\frac{\partial f}{\partial x}\right)$$

Sustituyendo este θ en la fórmula de la derivada parcial tendremos que el máximo cambio vale

$$M = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Las expresiones anteriores tienen operaciones de raíces cuadradas y cuadrados que pueden ser computacionalmente lentos. Por ello, a veces se realizan aproximaciones de estas funciones que son computacionalmente menos costosas.

7.2. Pasos en la detección de Bordes

Los algoritmos para la detección de bordes tienen tres pasos:

1. **Filtrado.** Puesto que el cálculo del gradiente basado en los valores de intensidad de un número finito de puntos es muy susceptible al ruido, normalmente se filtra la imagen inicialmente para mejorar el rendimiento del detector con respecto al ruido. Es obvio que este proceso ha de aplicarse con cuidado, puesto que el emborronamiento oculta los bordes.
2. **Realce.** Este paso es esencial para determinar cambios en la intensidad de los vecinos de un punto. El proceso es básicamente el cálculo del gradiente.
3. **Detección.** Muchos puntos en una imagen tienen un gradiente no nulo. Obviamente no todos ellos son bordes para todas las aplicaciones. Es necesario algún método para determinar qué puntos son bordes. Un criterio ampliamente usado es la umbralización.
4. **Estimación.** Esta deberá llevarse a cabo en la resolución apropiada y en algunos casos será necesario también estimar la dirección del gradiente.

7.3. Aproximaciones Discretas del Gradiente

En las dos últimas décadas se han desarrollado muchos detectores de fronteras, discutiremos aquí algunos de los más comunes. Es fundamental entender ahora que puesto que de lo que se dispone es una imagen y no la escena real, hemos de hacer aproximaciones de las derivadas basadas en diferencias finitas.

7.3.1. Operador de Roberts

El operador de Roberts es uno de los más antiguos. Es muy fácil de calcular ya que usa sólo una ventana de tamaño 2×2 . Sus máscaras de convolución son

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

la magnitud del gradiente se calcula entonces mediante la siguiente aproximación

$$G(f(i, j)) = |f(i, j) - f(i + 1, j + 1)| + |f(i, j + 1) - f(i + 1, j)|$$

puesto que G_x y G_y son aproximaciones de las derivadas parciales con respecto a x e y respectivamente.

Una gran desventaja del operador de Roberts es que es muy sensible al ruido, ya que se usan muy pocos píxeles para aproximar el gradiente, además, la aproximación está desplazada del punto (i, j) al $(i + 1/2, j + 1/2)$.

7.3.2. Operador de Sobel

Una forma de evitar que el gradiente se calcule en un punto intermedio, tal y como ocurre en el operador de Roberts, es usar entornos de tamaño 3×3 . Consideremos la distribución de píxeles que se muestra alrededor del (i, j) en la figura 7.3. El operador de Sobel es la magnitud del gradiente calculado mediante

$$M = \sqrt{s_x^2 + s_y^2}$$

donde las derivadas parciales se calculan mediante

$$\begin{aligned} s_x &= (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \\ s_y &= (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \end{aligned}$$

con la constante $c = 2$.

Es claro que estos operadores de gradiente pueden implementarse utilizando máscaras de convolución.

Es importante notar que este operador le da más peso a los píxeles más cercanos al centro de la máscara. Este operador es una de los más usados.

a_0	a_1	a_2
a_7	(i, j)	a_3
a_6	a_5	a_4

Figura 7.3: Etiquetado de niveles de gris para explicar los detectores de bordes

7.3.3. Operador de Prewitt

Este operador es similar al de Sobel, reemplazando la constante $c = 1$.

7.3.4. Otros Operadores de Derivadas

Aunque la formulación dada aquí es la más común para los operadores de fronteras existen monografías donde la formulación de los operadores de Sobel y Prewitt es diferente además de incluir otros operadores como los de Robinson y Kirsch, ver [48].

7.4. Operadores de Segunda Derivada

Los detectores de fronteras que hemos discutido calculan el gradiente y si su valor está por encima de un umbral, se supone la existencia de un punto de borde. El resultado es demasiados puntos de borde. Una mejor aproximación sería encontrar sólo los puntos que tienen máximos locales en el gradiente y considerar estos los puntos de borde. Esto significa que en los puntos de borde existe un pico en la primera derivada y equivalentemente un cruce por cero en la segunda. Así pues los puntos de borde se detectan encontrando los cruces por cero de la segunda derivada en las intensidades de la imagen.

Existen dos operadores bidimensionales que corresponden a la segunda derivada: el Laplaciano y la Segunda Derivada Direccional. Aquí analizaremos sólo el primero.

7.4.1. Operador Laplaciano

La laplaciana es el equivalente bidimensional de la segunda derivada. La fórmula para la laplaciana de una función $f(x, y)$ es

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

La segunda derivada en las direcciones x e y se aproximan utilizando las ecuaciones en diferencias

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{G_x}{\partial x} \\ &= \frac{\partial(f(i+1, j) - f(i, j))}{\partial x} \\ &= f(i+2, j) - 2f(i+1, j) + f(i, j)\end{aligned}$$

Análogamente obtendríamos

$$\frac{\partial^2 f}{\partial y^2} = f(i, j+2) - 2f(i, j+1) + f(i, j)$$

Sin embargo, estas dos aproximaciones no están centradas en la posición (i, j) de modo que una aproximación más razonable sería

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= f(i+1, j) - 2f(i, j) + f(i-1, j) \\ \frac{\partial^2 f}{\partial y^2} &= f(i, j+2) - 2f(i, j+1) + f(i, j)\end{aligned}$$

Combinando estas dos ecuaciones en un único operador, podemos usar la siguiente máscara para aproximar el laplaciano.

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Algunas veces es deseable darle más peso a los píxeles centrales del entorno. Una aproximación de la Laplaciana que hace esto es

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array}$$

El operador laplaciano marca la presencia de un borde cuando la salida del operador realiza una transición por cero. Los ceros triviales, es decir, las regiones de cero uniforme tienen que ser ignoradas. En principio, las localizaciones de los cruces por cero pueden ser estimadas a nivel de subpíxeles usando interpolación lineal; pero los resultados pueden ser inexactos.

2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8

0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0
0	0	0	6	-6	0	0	0

Figura 7.4: Salto y Resultado de Aplicar el Laplaciano

Consideremos el ejemplo que se muestra en la figura 7.4. Esta figura muestra el resultado de aplicar la laplaciana a una imagen de un paso salto simple. En este ejemplo, el cruce por cero que corresponde al borde en la imagen, está a mitad del camino de los dos píxeles con valores no nulos de la laplaciana. En este caso los bordes deberían ser marcados o bien a la izquierda o la derecha, pero de una forma consistente en toda la imagen.

Consideremos el ejemplo de la figura 7.5. Esta contiene la respuesta a una rampa. Este cruce por cero corresponde a un píxel en la imagen. Obviamente ésta es una situación ideal y la localización de los bordes debería determinarse por interpolación.

7.4.2. Cruces Por cero de la segunda Derivada. Laplaciana de Gaussiana

En los años setenta, la teoría de Marr (ver capítulo sobre Ciencias Cognitivas) concluyó a partir de experimentos neurofisiológicos que las fronteras de los objetos son los datos más importantes para unir una imagen de intensidades con su interpretación. Los detectores de bordes que existían en aquel tiempo estaban basadas en la convolución con máscaras muy pequeñas y sólo funcionaban sobre determinadas imágenes. Su principal desventaja era su dependencia de los tamaños de los objetos y su sensibilidad al ruido.

2	2	2	2	2	5	8	8	8	8
2	2	2	2	2	5	8	8	8	8
2	2	2	2	2	5	8	8	8	8
2	2	2	2	2	5	8	8	8	8
2	2	2	2	2	5	8	8	8	8
2	2	2	2	2	5	8	8	8	8

0	0	0	3	0	-3	0	0
0	0	0	3	0	-3	0	0
0	0	0	3	0	-3	0	0
0	0	0	3	0	-3	0	0

Figura 7.5: Otro Salto y Resultado de Aplicar el Laplaciano

Una técnica de detección de bordes basada en los cruces por cero de la segunda derivada (en su forma original el detector de Marr y Hildreth, [29]) explora el hecho de que un borde de salto corresponde a un cambio brusco de la función de la imagen. La primera derivada de la función de la imagen debe tener un extremo en la posición correspondiente a un borde en la imagen, y la segunda derivada debe ser cero en la misma posición. Sin embargo, es mucho más fácil y preciso encontrar un cruce por cero que un extremo. (ver figura 7.6). En esta figura se ilustra el problema en el caso unidimensional.

El punto crucial es cómo calcular la segunda derivada de una forma robusta. Una po-

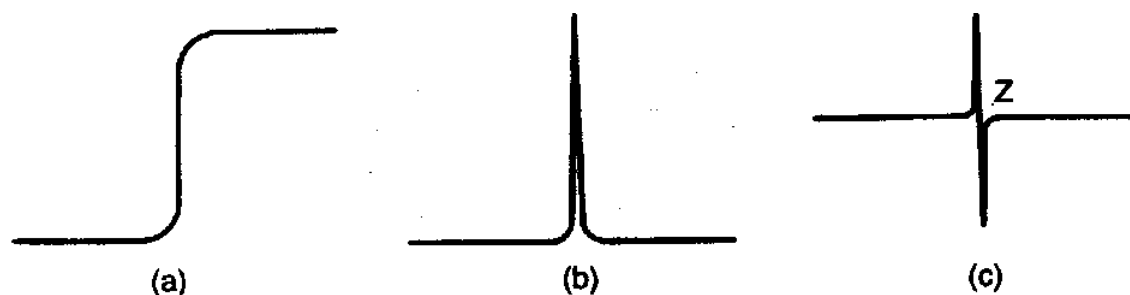


Figura 7.6: (a) Intensidad, (b) Derivada Primera (c) Derivada segunda

sibilidad es suavizar la imagen primero (para reducir ruido) y después calcular la segunda derivada. Cuando se selecciona un filtro de alisamiento se deben cumplir dos criterios: debe ser suave y de banda limitada para reducir el número de frecuencias en las que puede producirse el cambio y segundo que la respuesta sea casi puntual. Aunque estos dos criterios entran en conflicto, la distribución que optimiza simultáneamente ambos es la distribución gaussiana.

El filtro de alisamiento gaussiano $2D$, $G(x, y)$ viene dado por

$$G(x, y) = c \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right]$$

donde x e y son las coordenadas de la imagen, σ es la desviación típica de la distribución de probabilidad y c es una constante de normalización.

Nuestro objetivo es obtener la segunda derivada de una versión suave, de la imagen original $f(x, y)$. Ya hemos visto que el operador laplaciano ∇^2 produce la segunda derivada y es además isotrópico. Consideremos entonces la laplaciana de una imagen $f(x, y)$ suavizada por una gaussiana. Tendremos entonces que calcular

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

El orden en el que se realiza la convolución y diferenciación puede ser intercambiado debido a la linealidad de los operadores y tendremos

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

si notamos el filtro de convolución por $h(x, y, \sigma)$ tendremos que

$$h(x, y, \sigma) = \frac{c}{\sigma^2} \left(\frac{x^2 + y^2}{\sigma^2} - 2 \right) \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right]$$

que recibe el nombre de filtro del sombrero mejicano.

Este proceso de búsqueda de la segunda derivada es muy robusto. El alisamiento gaussiano suprime al influencia de píxeles lejanos y la laplaciana es una medida eficiente y estable de los cambios en una imagen.

La localización en la imagen $\nabla^2 G$ donde se cruza por cero corresponde a la posición de los bordes. La ventaja de esta aproximación comparada con los clásicos operadores de bordes es que una mayor área que rodea el píxel se tiene en cuenta. La influencia de los píxeles distantes está determinada por σ .

Experimentos neurofisiológicos proporcionan evidencia de que la retina del ojo humano en la forma de células ganglionares realiza operaciones similares a la Laplaciana de Gaussiana (**LoG**). Cada célula responde al estímulo de luz en un entorno local llamado el campo receptivo que tiene una estructura redondeada de dos tipos complementarios (fuera-centro y dentro-centro). Cuando un estímulo de luz ocurre, la actividad de las células dentro-centro aumenta y la de las fuera-centro se inhibe. Este proceso puede describirse como la convolución de una imagen con un operador **LoG**.

El operador **LoG** puede ser aproximado de un modo efectivo por la convolución con una máscara que sea la diferencia de dos gaussianas, con valores muy diferentes de σ . Este método recibe el nombre de **Diferencia de Gaussianas** también notado **DoG**.

Esta aproximación tradicional al cruce por cero tiene también inconvenientes. Primero, suaviza mucho la forma, además de crear contornos cerrados en forma de spaghetti.

7.5. Detectores de Bordes Gaussianos

La idea esencial en la detección de bordes de salto es encontrar puntos en la imagen que tengan localmente máximo gradiente. Muchas de las investigaciones en la detección de bordes está dedicada a la aproximación numérica del gradiente en imágenes. Los saltos en las imágenes reales no son abruptos ya que éstos son suavizados por los filtros de paso bajo inherentes a los procesos de captación de imágenes. Análogamente las imágenes son distorsionadas por el ruido. Por tanto, cualquier proceso de detección de bordes basado en fronteras debe por una parte suprimir el ruido y por otra localizar los bordes tan preciso como sea posible. Sin embargo, como ya hemos visto anteriormente estos dos requerimientos entran en conflicto. De nuevo la mejor aproximación basada en gradientes hace uso de gaussianas. Sin consideramos un operador que sea la combinación de un filtro gaussiano con una aproximación de gradiente tendremos un operador que es sensible al borde en la dirección de máximo cambio. Dentro de este modelo se encuadra el detector de Canny (ver [6]) que veremos a continuación.

7.5.1. El detector de Bordes de Canny

El detector de bordes de Canny es la primera derivada de una gaussiana y aproxima muy bien el operador que optimiza el producto del cociente señal ruido y localización. El algoritmo se resume como sigue: Sea $f(x, y)$ la imagen. El resultado de convolucionar la

imagen con un filtro gaussiano es la imagen suavizada

$$s(x, y) = G(x, y, \sigma) * f(x, y)$$

donde σ es la anchura de la gaussiana y controla el grado de alisamiento. A continuación se calcula el gradiente de la imagen suavizada usando una aproximación 2×2 de las primeras diferencias y que producen las matrices $P(i, j)$ y $Q(i, j)$ para las derivadas parciales en x e y

$$P(i, j) \approx (s(i+1, j) - s(i, j) + s(i+1, j+1) - s(i, j+1))/2$$

$$Q(i, j) \approx (s(i, j) - s(i, j-1) + s(i+1, j) - s(i+1, j-1))/2$$

Las diferencias finitas son promediadas sobre un cuadrado 2×2 de forma que las derivadas parciales se calculan en el mismo punto de la imagen. La magnitud y orientación del gradiente se puede calcular mediante

$$M(i, j) = \sqrt{P(i, j)^2 + Q(i, j)^2}$$

$$\theta(i, j) = \arctang(Q(i, j), P(i, j))$$

donde la función arcotangente toma dos argumentos y genera un ángulo sobre el círculo completo de direcciones posibles.

La matriz de magnitudes $M(i, j)$ tendrá valores grandes donde el gradiente sea grande, pero esto no es suficiente para identificar los bordes ya que el problema de encontrar los puntos donde hay un cambio rápido de la intensidad se ha transformado en la búsqueda de los puntos donde existe un máximo local de $M(i, j)$. Para identificar los bordes, las crestas en la matriz de magnitudes debe reducirse de forma que las magnitudes donde existe un máximo local sean las únicas que persistan.

Dos son las aproximaciones que vamos a examinar para reducir las crestas. Una vez calculada la magnitud del gradiente y su dirección, estos valores son examinados en la imagen completa junto con la magnitud de los bordes de los dos píxeles vecinos en la dirección perpendicular al borde. Si al menos uno de estos píxeles tiene magnitud superior que la del pixel central, a este último se le asigna el valor cero, es decir $M(i, j) = 0$. Este proceso hace más estrecha las crestas. Los valores de las alturas de las crestas se retienen en la magnitud de no supresión que es notada $N(i, j)$. Observemos que $N(i, j)$ vale $M(i, j)$ o cero si no lo hemos localizado como máximo local. Los valores no nulos en $N(i, j)$ corresponden a la cantidad de contraste en un salto en la imagen. Una aproximación discreta de este método puede encontrarse descrita en [48].

A pesar del alisamiento realizado previamente, esta nueva matriz contendrá muchos falsos bordes. Un procedimiento típico que se usa para reducir el número de fragmentos de falsos bordes es aplicar un umbral a $N(i, j)$. Todos los valores por debajo del umbral se cambian a cero. Todavía existirán algunos bordes falsos por ser el umbral muy bajo τ (falsos positivos) y partes del contorno no aparecerán (falsos negativos). Para resolver este problema se usa un umbral doble. A $N(i, j)$ le aplicamos dos umbrales τ_1 y τ_2 con $\tau_2 \approx 2\tau_1$ y tenemos dos imágenes umbralizadas $T_1(i, j)$ y $T_2(i, j)$, respectivamente. Puesto que la imagen T_2 se formó con un umbral superior ésta tendrá menor número de falsos positivos, pero también tendrá numerosos huecos en los contornos (falsos negativos). El algoritmo de doble umbral une los bordes de T_2 para formar contornos. Cuando llega al final de un contorno, el algoritmo mira a T_1 , mira a los 8 vecinos del píxeles que pueden ser unidos al contorno. El algoritmo continúa utilizando píxeles de T_1 hasta llegar a otro píxel en T_2 .

7.6. Medidas de la Calidad de un detector de bordes

La idea de medir la calidad de un detector de bordes es bastante antigua y Abdou y Pratt ([2]) lo han usado para definir el mérito de cada detector. La idea no es usar el mérito para definir el detector (en la línea de lo hecho con el filtro de Canny) sino usar el mérito para seleccionar el detector. La figura 7.7 muestra unos cuantos defectos que puede tener un detector. El mérito definido por Abdou y Pratt es

$$F = \frac{1}{\text{máx}(I_I, I_A)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d^2(i)}$$

donde I_I es el número ideal de puntos de borde, I_A es el número de puntos de borde detectados, $d(i)$ es la menor distancia del borde i -ésimo al borde ideal y α es una constante positiva. F es siempre menor o igual que uno y alcanza la igualdad cuando $I_I = I_A$ y $d(i) = 0$ para todo i

7.7. Técnicas Jerárquicas

La mayor parte del trabajo relacionado con la búsqueda de bordes en una imagen se debe a la enorme cantidad de píxeles que hay que examinar. La mayor parte de este trabajo no es necesario, el ser humano puede observar rápidamente que áreas de la imagen son

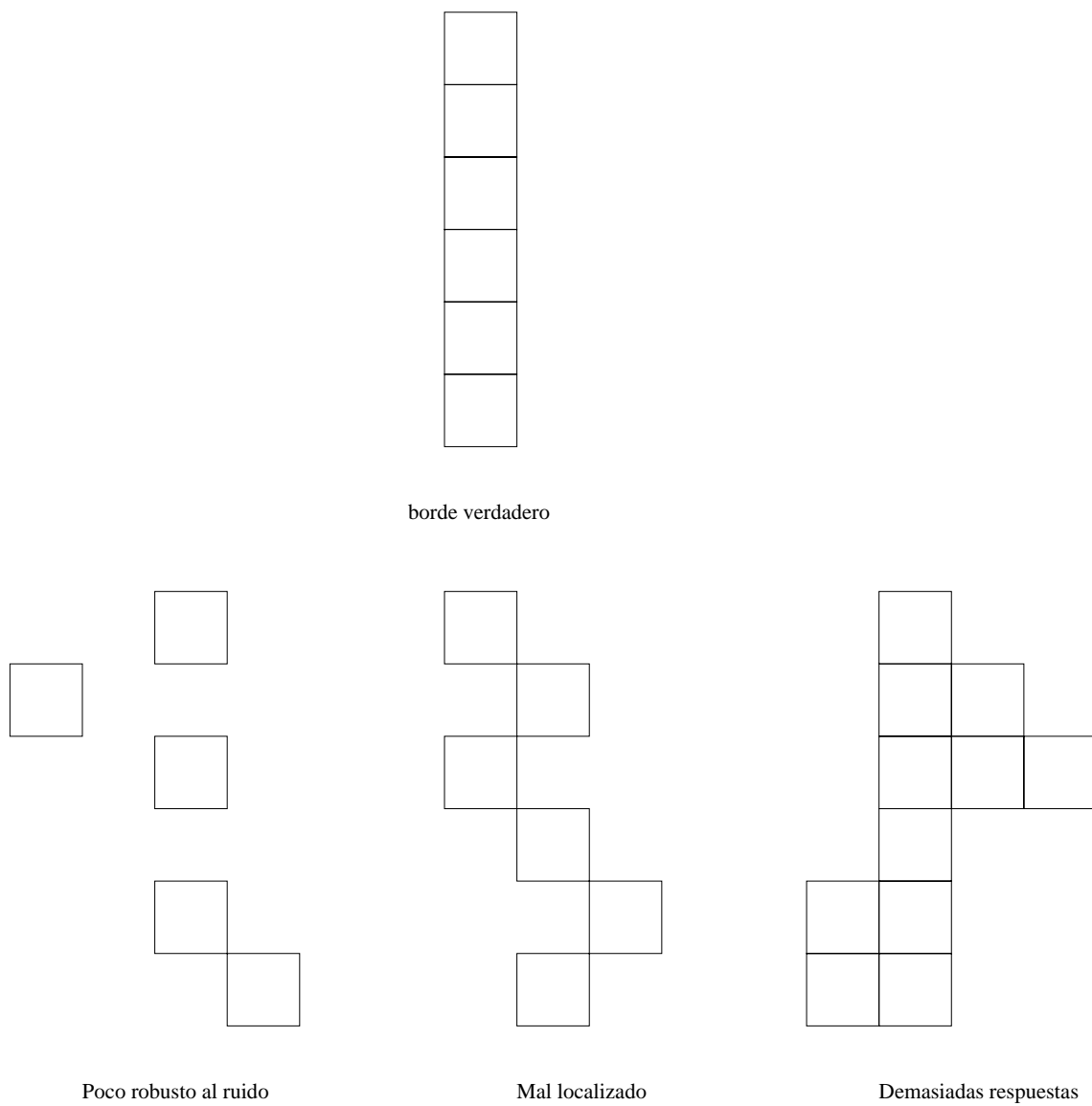


Figura 7.7: Defectos de Bordes

interiores de regiones y por tanto no contendrán fronteras, con esta información poco fina el cerebro parece entonces capaz de aplicar una resolución superior a regiones específicas de interés para separar las regiones con más precisión. Las técnicas jerárquicas tratan de emular esta conducta sacrificando resolución en pasos iniciales para tener una idea de la región donde se encuentra la frontera.

Supongamos que tenemos una imagen cuadrada I cuya longitud de cada lado es una potencia de dos, $N = 2^n$, podemos entonces definir imágenes recursivamente como sigue

$$I_n = I$$

$$I_{i-1}(x, y) = [I_i(2x, 2y) + I_i(2x + 1, 2y) + I_i(2x, 2y + 1) + I_i(2x + 1, 2y + 1)]/4$$

$$0 \leq x, y < 2^{i-1}$$

así pues para $0 \leq i \leq n$, I_i es una imagen de tamaño $2^i \times 2^i$ con niveles de gris en el mismo rango que I ; en particular I_0 es una imagen con un único píxel con nivel de gris la media de los niveles de gris de la imagen original.

Es muy importante tener en cuenta que el tamaño en píxeles que se necesita para almacenar esta estructura jerárquica es

$$\sum_{i=0}^n 2^i \times 2^i = \frac{44^n - 1}{3} \approx \frac{4}{3}(\text{número de píxeles en } I)$$

por tanto, a pesar de generarse una enorme cantidad de imágenes, la carga de almacenamiento no es prohibitiva en comparación con lo que se requiere para la imagen I .

Las imágenes intermedias sufrirán una creciente cantidad de emborronamiento al reducirse su resolución. Mientras que se pierden detalles, esto no importa si estamos sólo buscando los rasgos más importantes, o bien su naturaleza aproximada.

El hecho de que las regiones principales suelen ser distinguibles también a resoluciones inferiores se explota en el siguiente algoritmo de segmentación, ([49]). Elegimos el nivel más alto de la pirámide (es decir, el valor menor de i) en el que los rasgos que queremos extraer están todavía presentes (la elección de este nivel no es necesariamente una tarea fácil) y le pasamos un detector de fronteras. En cada píxel en el que el operador produce la evidencia de la existencia de un borde, descendemos la pirámide una capa y pasamos el detector de bordes sobre los cuatro píxeles que se usaron para formarlo. Este proceso se ejecuta recursivamente hasta que se alcanza $I_n = I$, lo que significa que hemos aplicado el detector de bordes a los píxeles de la imagen. Es muy importante tener en cuenta que el detector de bordes sólo se aplica a los píxeles de la imagen a los que se ha llegado de esta forma.

Esta idea tiene dos ventajas sobre la aplicación de los detectores de bordes sólo a la imagen original; primero el proceso de promedio tiene como efecto distinguir las regiones entre sí y también del fondo, aunque se emborronen los bordes. Segundo, es computacionalmente más económico; usualmente este algoritmo comenzará a un tercio de la imagen original, es decir con $2^{4n/3}$ píxeles en lugar de 2^{2n} , y el operador de bordes será aplicado en los niveles más bajos donde sea necesario.

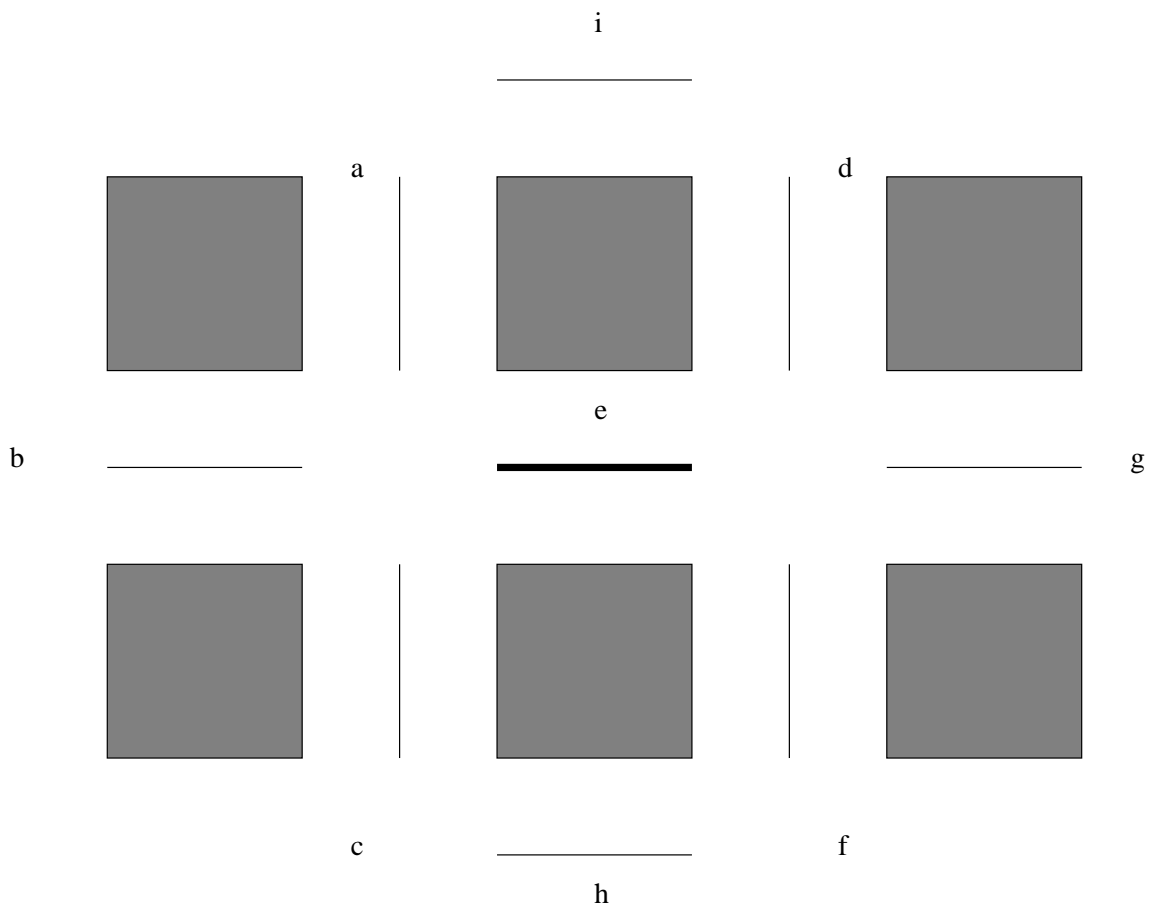
7.8. Técnicas de Segmentación Basadas en Bordes

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

Existen diferentes métodos para la segmentación basada en bordes que difieren en la estrategia para la construcción de la frontera, así como en la cantidad de información previa que se incorpora en el método. Es obvio, que a mayor información mejor segmentación. En este capítulo discutiremos sólo un método que incorpora muy poca información en el proceso de segmentación. Métodos más complejos y también más efectivos son discutidos en [48].

7.8.1. Relajación de Bordes

Las fronteras que se obtienen de los métodos descritos en las secciones anteriores están usualmente muy afectados tanto por el ruido de la imagen como por la no presencia de partes que deberían estar presentes. Si en el proceso de paso de borde a frontera se tiene en cuenta las propiedades de los bordes vecinos presentes, es posible aumentar la calidad de los resultados. Todas las propiedades de una imagen junto con la posibilidad de existencia de más bordes, se evalúan iterativamente con más precisión hasta que el contexto de los bordes queda claro. Basado en la fuerza de bordes en una vecindad local especificada,

Figura 7.8: Borde tipo grieta e con sus vecinos

la confianza de cada borde se aumenta o disminuye ([1],[52],[45],[15]). Un borde débil posicionado entre dos bordes fuertes proporciona un ejemplo de contexto; es altamente probable que este borde débil entre ellos sea parte de una frontera. Si, por otra parte, un borde, incluso uno fuerte no tiene contexto que lo soporte, puede que no sea parte de ninguna frontera.

El método que vamos a discutir aquí ([16], [41]) es un ejemplo clásico de evaluación de contexto en bordes. El método utiliza bordes tipo grieta (entre píxeles). El contexto del borde se considera a ambos lados de un borde, e , con el sistema de vecinos que se muestra en la figura 7.8.

Todas las posibles posiciones de los vecinos de e deben ser incluidas para cubrir la

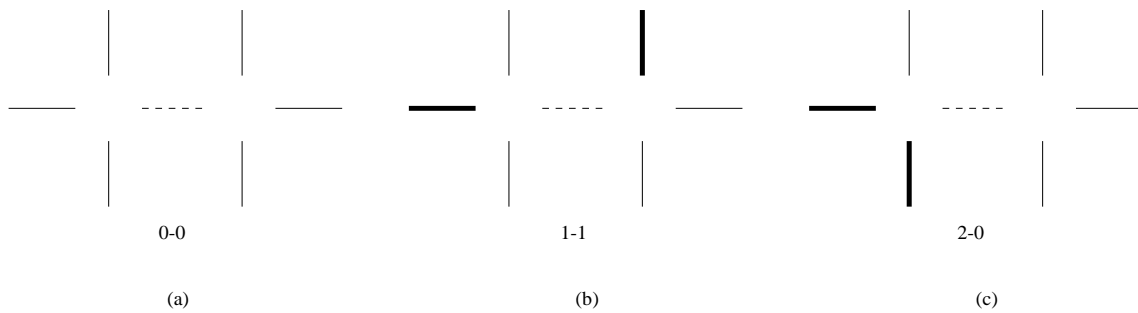
forma en la que se continua en el entorno. Además dos posiciones paralelas con e pueden ser incluidas en el sistema de vecinos -estas posiciones paralelas compiten con e en la colocación del borde. La relajación de bordes busca la construcción de fronteras continuas de forma que discutiremos los patrones que podemos encontrar en el entorno. Dado un borde central e , éste tiene un vértice a cada uno de sus lados y cada uno de estos vértices tiene tres posibles formas de continuar. La forma de continuar de un vértice se evalúa de acuerdo al número de bordes que de él emanan, sin contar e , y este número define el tipo de vértice. Entonces el tipo de borde e puede ser representado mediante $i - j$, donde i notará la configuración de bordes que se produce a su izquierda (en su vértice izquierdo) y j el que hay a la derecha. Por ejemplo, asignaremos el tipo $0 - 0$ a la configuración que se muestra en la figura 7.9(a), el $2 - 0$ para la figura 7.9(c), etc. Es obvio que por simetría debemos considerar sólo el caso $i \leq j$. Tenemos las siguientes situaciones de contexto:

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

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

El algoritmo funcionaría como sigue: **Relajación de bordes**

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

Figura 7.9: Borde tipo grieta e con sus vecinos

2. encontrar el tipo de borde de cada borde basado en la confianza de los bordes de sus vecinos,
3. Actualizar la confianza $c^{k+1}(e)$ de cada borde e de acuerdo con su tipo y su confianza previa,
4. Parar si todas las confianzas de los bordes han tendido a cero o uno. Repetir los pasos (2)-(3) en caso contrario.

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

$$\begin{aligned} \text{tipo}(i) &= \max_k(\text{tipo}(k)), k = 0, 1, 2, 3 \\ \text{tipo}(0) &= (m - a)(m - b)(m - c) \\ \text{tipo}(1) &= a(m - b)(m - c) \\ \text{tipo}(2) &= ab(m - c) \\ \text{tipo}(3) &= abc \end{aligned}$$

donde a, b, c son los valores normalizados de los otros bordes y sin pérdida de generalidad podemos suponer que $a \geq b \geq c$, q es una constante para la que un valor de 0,1 parece apropiado y $m = \max(a, b, c, q)$. Observemos que la introducción de la cantidad q asegura que $\text{tipo}(0)$ es no cero para pequeños valores de a .

Por ejemplo si elegimos $q = 0,1$ un vértice $(a, b, c) = (0,5, 0,05, 0,05)$ es un vértice de tipo uno, mientras que un vértice $(0,3, 0,2, 0,2)$ es un vértice tipo tres. Los mismos

resultados se pueden obtener observando el número de bordes de un vértice que están por encima de un umbral. El tipo de borde se encuentra como la concatenación del tipo de vértices y la confianza de bordes se modifica como sigue

$$\begin{aligned} \text{Aumento de confianza} \quad c^{k+1}(e) &= \min(1, c^k(e) + \delta) \\ \text{Disminución de confianza} \quad c^{k+1}(e) &= \max(0, c^k(e) - \delta) \end{aligned}$$

donde δ es una constante apropiada, usualmente en el rango $[0,1,0,3]$.

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

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

$$\text{si } c^{k+1}(e) > T_1 \quad \text{asignar entonces } c^{k+1}(e) = 1 \quad (7.1)$$

$$\text{si } c^{k+1}(e) < T_2 \quad \text{asignar entonces } c^{k+1}(e) = 0 \quad (7.2)$$

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

7.9. Ejercicios

1. Implementar los detectores de bordes basados en los operadores de Roberts, Sobel y Prewitt.
2. Implementar el detector de bordes basado en la laplaciana de la gaussiana.
3. Opcional: Implementar el detector de Canny.
4. Implementar un detector de bordes jerárquico.
5. Opcional: Implementar una técnica de segmentación basada en bordes.

Capítulo 8

Segmentación de Imágenes

8.1. Introducción

En otros temas hemos estudiado además de material de base, algunas técnicas de mejora de imágenes, que en algunos casos reciben el nombre de técnicas de preprocesamiento. Nos proponemos ahora avanzar en la línea de extraer información de una imagen, este área recibe el nombre de *análisis de imágenes* ([13],[48],[43]). La extracción de información podría decirse que ya comenzó a hacerse en el tema de segmentación basada en fronteras cuando vimos las técnicas de relajación. Sin embargo, la búsqueda de regiones, bien por homogeneidad o disparidad se hará más exhaustivamente aquí.

El primer paso en el análisis de imágenes es generalmente segmentar la imagen en regiones. *Una región, en una imagen, es un grupo de píxeles conectados (un concepto que necesita ser definido precisamente) que tienen propiedades similares.* Es obvio que las regiones son importantes para la interpretación de las imágenes pues pueden corresponder a objetos en la escena. Una imagen puede contener varios objetos y además cada objeto puede contener varias regiones que corresponden a partes del mismo.

Para que una imagen pueda ser interpretada correctamente, tiene que ser dividida en regiones que correspondan a objetos o partes de ellos. Sin embargo, debido a los errores de segmentación, la correspondencia entre regiones y objetos no será perfecta y deberemos utilizar conocimiento específico en etapas posteriores de la interpretación de las imágenes.

8.2. Regiones y Bordes

Consideremos una imagen simple que contenga diferentes objetos. El primer paso en el análisis para *entender* una imagen es particionar la imagen de forma que las regiones que representan diferentes objetos son marcadas explícitamente. Estas particiones pueden ser obtenidas a partir de las características de los niveles de gris en la imagen, o utilizando alguna otra. Así pues, podría decirse que los valores de gris en las posiciones de la imagen son las observaciones y que los otros atributos, como la pertenencia a regiones, deben obtenerse a partir de los niveles de gris. Existen dos aproximaciones para particionar una imagen en regiones: *segmentación basada en regiones* y *estimación de la frontera utilizando detectores de bordes*.

En la aproximación basada en regiones todos los píxeles que corresponden a un objeto se agrupan juntos y son marcados para indicar que pertenecen a una región. Este proceso recibe el nombre de *segmentación*. Los píxeles son asignados a regiones según algún criterio que los distingue del resto de la imagen. Dos principios muy importantes en la segmentación son valor de similitud y proximidad espacial. Dos píxeles pueden ser asignados a la misma región si tienen características similares de intensidad y si están próximos. La varianza de los niveles de gris en una región y la compactificación de una región pueden ser también usadas como valores de similitud y proximidad entre píxeles respectivamente.

Los principios de similitud y proximidad aparecen por la hipótesis de que los puntos de un mismo objeto producirán en la imagen píxeles con niveles de gris similares y próximos espacialmente. Obviamente, esto no es cierto en muchas situaciones. Podemos, sin embargo, agrupar píxeles en una imagen utilizando estos principios y usar conocimiento que depende del dominio para realizar la correspondencia objeto-región.

Al igual que hemos abordado el problema de la segmentación por similitud entre los niveles, es posible también realizar este proceso mediante lo que se denomina *disparidad*. El objetivo es encontrar los píxeles que se encuentran en las fronteras de las regiones. Estos píxeles, llamados bordes, pueden localizarse examinando los píxeles vecinos. Puesto que los píxeles llamados bordes están en la frontera de las regiones y éstas suelen tener diferentes niveles de gris a cada lado, lo que necesitamos es medir las diferencias entre los píxeles vecinos. La mayoría de los detectores de bordes usan sólo las características de la intensidad para su detección, aunque algunos aspectos como la textura y el movimiento también aportan información. Observemos, por ejemplo, que en una segmentación para

detectar objetos en movimiento la comparación se haría entre lo observado en dos imágenes consecutivas.

En imágenes ideales, una región estará acotada por un contorno cerrado. En principio, la segmentación de regiones y detección de bordes debería producir los mismos resultados. Las fronteras (cerradas) podrían obtenerse entonces a partir de las regiones utilizando un algoritmo de seguimiento de fronteras. Análogamente, las regiones pueden obtenerse a partir de los bordes usando un algoritmo de relleno. Por desgracia, en las imágenes reales es raro obtener tanto bordes a partir de regiones como regiones a partir de bordes. Debido al ruido y otros factores ni la segmentación de regiones ni la detección de bordes proporcionan información perfecta.

En este tema vamos a abordar la segmentación de imágenes desde los puntos de vista de la similitud y disparidad, aunque desde este punto de vista algo fue analizado en el tema anterior. Sin embargo, realizaremos una subdivisión adicional, los métodos de segmentación serán divididos en tres grupos en función de los rasgos dominantes que emplean. Primero *Conocimiento global* sobre la imagen o sus partes; este conocimiento se basa principalmente en el histograma de los rasgos, para continuar con la *segmentación basada en bordes* y la *segmentación basada en regiones*.

Antes de pasar a estudiar estos modelos de segmentación, definamos precisamente que entenderemos por segmentación. Una segmentación completa de una imagen R es un conjunto finito de regiones R_1, \dots, R_s tales que

$$R = \cup_{i=1}^s R_i \quad R_i \cap R_j = \emptyset \quad i \neq j$$

8.3. Segmentación Basada en Umbralización

Consideremos el histograma de niveles de gris mostrado en la figura 8.1 correspondiente a una imagen $f(x, y)$ compuesta de objetos oscuros sobre fondo luminoso, de forma que los píxeles en los objetos y fondo tienen niveles de gris agrupados en dos modas dominantes. Una forma obvia de extraer los objetos del fondo es seleccionar un umbral T que separe estas modas. Es decir, escribir

$$g(i, j) = \begin{cases} 1 & \text{si } f(i, j) \geq T \\ 0 & \text{si } f(i, j) < T \end{cases}$$

Esta ecuación tiene muchas modificaciones. Una posibilidad es segmentar la imagen en regiones de píxeles con nivel de gris en un conjunto D y el resto como fondo. También

podemos utilizar umbrales múltiples, lo que produce imágenes no binarias con un conjunto muy limitado de niveles de gris, el proceso sería

$$\begin{aligned}
 g(i, j) &= 1 && \text{si } f(i, j) \in D_1 \\
 &= 2 && \text{si } f(i, j) \in D_2 \\
 &= 3 && \text{si } f(i, j) \in D_3 \\
 &\dots \\
 &= n && \text{si } f(i, j) \in D_n \\
 &= 0 && \text{en el resto}
 \end{aligned}$$

Este tipo de umbralización multinivel es generalmente menos fiable que la umbralización en dos regiones. La razón está en que es difícil establecer múltiples umbrales que aislen regiones correctamente, especialmente cuando hay muchas modas (máximos locales).

Antes de seguir con el análisis de los métodos de umbralización, es importante tener en cuenta el papel de la iluminación en la segmentación, aunque no lo comentaremos aquí, el lector interesado puede consultar [13].

Es claro que para hacer la segmentación más robusta, el umbral debería ser seleccionado automáticamente por el sistema. Tanto el conocimiento sobre la escena como el del problema a resolver y cualquier otro conocimiento debería ser usado para fijar el umbral. La información de la que podemos disponer puede venir proporcionada por las características de los objetos, el tamaño de los mismos, la fracción de la imagen ocupada por los objetos y los diferentes tipos de objetos que pueden aparecer en la imagen.

8.3.1. Método P-cuantil

Este método usa conocimiento sobre el área o el tamaño del objeto que se desea extraer para segmentar la imagen. Supongamos que en una aplicación determinada los objetos ocupan alrededor del p por ciento del tamaño del área de la imagen. Utilizando este conocimiento es posible particionar la imagen de modo que asigne el p por ciento de los píxeles al objeto.

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

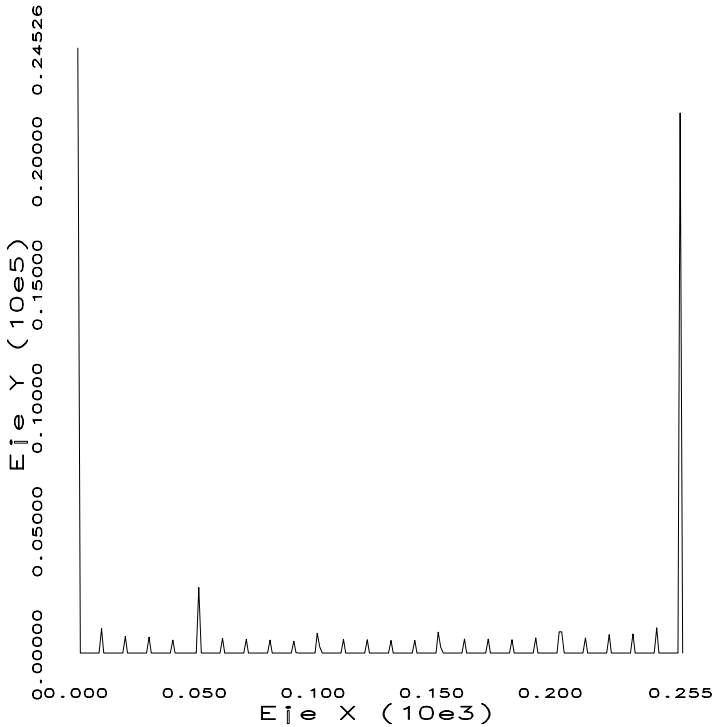


Figura 8.1: Imagen original y su histograma.

8.3.2. Método de la Moda

Supongamos que una imagen contiene sólo dos regiones con diferente nivel de gris. El histograma de tal imagen puede considerarse como un estimador de la densidad de probabilidad del nivel de gris, $p(z)$. Esta densidad es la suma o mixtura de dos densidades unimodales, una para la región luminosa y otra para la región oscura. Además, los parámetros de mixtura son proporcionales al área de cada región. Si la forma de la densidad es conocida o se supone conocida, es posible determinar un umbral, en términos de error mínimo que segmente la imagen en las dos regiones.

Veamos el proceso, supongamos que una imagen contiene dos valores combinados con ruido gaussiano aditivo. La función de densidad de la mixtura es

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

Donde P_1 y P_2 cumplen las condiciones $P_1 + P_2 = 1$ y las distribuciones asociadas a $p_1(z)$ y $p_2(z)$ son $\mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, 2$, respectivamente. Si todos los parámetros son conocidos, podemos determinar el umbral fácilmente.

Supongamos que las regiones oscuras corresponden a fondo y las luminosas a objetos. En este caso, $\mu_1 < \mu_2$ y podremos encontrar un umbral T de modo que todos los píxeles con nivel de gris por debajo de él son considerados puntos de fondo y por encima puntos de objeto. Tendremos entonces que la probabilidad de clasificar erróneamente un punto de objeto como de fondo es

$$E_1(T) = \int_{-\infty}^T p_2(z) dz$$

y la probabilidad de clasificar erróneamente un punto de fondo como de objeto es

$$E_2(T) = \int_T^{\infty} p_1(z) dz$$

La probabilidad de error total es

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

Para encontrar el umbral para el que este valor es mínimo derivamos la ecuación anterior y tendremos que el umbral debe satisfacer

$$P_1 p_1(T) - P_2 p_2(T) = 0$$

Aplicando este resultado a la distribución gaussiana, tomando logaritmos, tendremos

$$\log P_1 - \frac{(T - \mu_1)^2}{2\sigma_1^2} - \log(\sigma_1) = \log P_2 - \frac{(T - \mu_2)^2}{2\sigma_2^2} - \log(\sigma_2)$$

que produce una ecuación cuadrática en T que hemos de resolver. Si por ejemplo las dos varianzas son iguales y las notamos σ^2 tendremos que

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \log(P_2/P_1)$$

En este caso si las probabilidades iniciales son iguales $P_1 = P_2$, el umbral óptimo es la media de las medias.

Es claro que para el análisis anterior necesitamos conocer los parámetros que caracterizan a las distribuciones, parámetros que en muchas ocasiones no son conocidos. Una posibilidad es usar el error cuadrático medio y minimizar

$$\sum_{i=1}^n [p(z_i) - h(z_i)]^2$$

donde la mixtura de las distribuciones normales se nota $p(z)$, el histograma observado es $h(z)$ y n es el número de puntos.

Junto a esta posibilidad de estimación de los parámetros, es también posible diseñar métodos iterativos. Una aproximación muy simple es la dada por el siguiente algoritmo para encontrar iterativamente el umbral.

Selección Iterativa del Umbral

1. Seleccionar un estimador inicial del umbral T . Un buen valor puede ser la media de la imagen.
2. Realizar la partición de la imagen utilizando este umbral T , obteniendo así las regiones R_1 y R_2 .
3. Calcular las medias de estas particiones, μ_1 y μ_2 , respectivamente.
4. Seleccionar un nuevo umbral $T = \frac{\mu_1 + \mu_2}{2}$
5. Repetir los pasos 2 a 4 hasta que las medias μ_1 y μ_2 no cambien.

8.3.3. Umbralización Adaptativa

Si la iluminación de la escena es desigual, entonces el método que acabamos de describir puede no ser apropiado. La iluminación desigual puede ser debida a sombras o la propia dirección de la iluminación. En todos estos casos, el mismo umbral no puede ser usado

en toda la imagen. Los métodos adaptativos examinan regiones pequeñas de la imagen y entonces analizan la subimagen para obtener un umbral de sólo esa región.

Una aproximación muy simple para segmentar este tipo de imágenes es particionar la imagen en $m \times m$ subimágenes y seleccionar un umbral para cada subimagen basado en su subimagen. La segmentación final de la imagen es la unión de las regiones de sus subimágenes.

8.3.4. Umbralización en Estructuras de Datos Jerárquicas

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

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

Aunque en esta sección hemos discutido las principales aproximaciones a la detección

de fronteras basada en la umbralización, existen muchas variaciones que no hemos mencionado. El lector interesado puede consultar principalmente los libros [48], [13] y [43]. El artículo [40] es una excelente recopilación de métodos de segmentación basados en umbralización.

8.3.5. Selección del Umbral Basada en Características de la Frontera

Uno de los aspectos más importantes de la selección del umbral es su capacidad para identificar de una forma consistente los picos del histograma. Esta característica es particularmente importante para la umbralización automática en aquellas situaciones en las que las características de la imagen pueden cambiar sobre un amplio rango de intensidades. Es claro que las posibilidades de seleccionar un buen umbral aumentan si los picos del histograma son altos, estrechos, simétricos y separados por valles profundos.

Una aproximación para mejorar la forma del histograma es considerar sólo aquellos píxeles que son parte de o están cerca de la frontera entre objetos y fondo. Aunque las ventajas de esta aproximación son obvias, es claro que su principal problema es la hipótesis implícita de que la frontera entre objetos y fondo es conocida.

8.4. Técnicas de Segmentación Basadas en Bordes

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

Existen diferentes métodos para la segmentación basada en bordes que difieren en la estrategia para la construcción de la frontera, así como en la cantidad de información previa que se incorpora en el método. Es obvio, que a mayor información mejor segmentación.

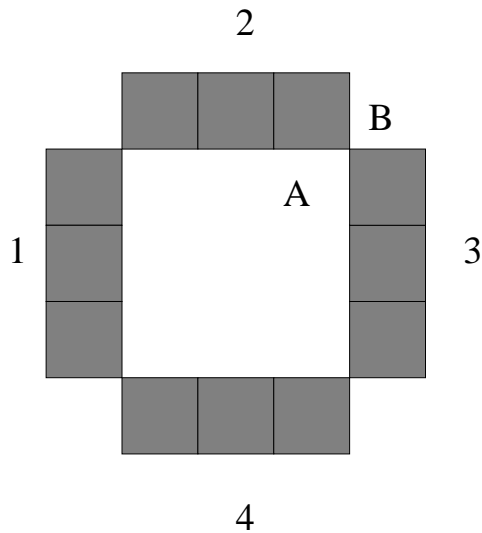


Figura 8.2: Problemas con la conectividad.

En este capítulo discutiremos sólo métodos que incorporan muy poca información en el proceso de segmentación. Métodos más complejos y también más efectivos son discutidos en [48].

8.4.1. Extracción de la Frontera

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

Conectividad

Conceptualmente, las fronteras pueden encontrarse trazando los bordes conectados (bien sea 4 u 8 conectados). Sin embargo, hay dificultades asociadas con estas definiciones de conectividad, como muestra la figura 8.2. Si utilizamos la 4-conectividad, los segmentos 1, 2, 3 y 4 serían clasificados como disjuntos, aunque obviamente parecen formar un anillo. Bajo la 8-conectividad estos segmentos estarían conectados, pero también lo estaría el interior con el exterior del anillo. En principio, esto podría resolverse utilizando tipos de conectividad distinta para los objetos y para el fondo, pero, obviamente hay otros problemas.

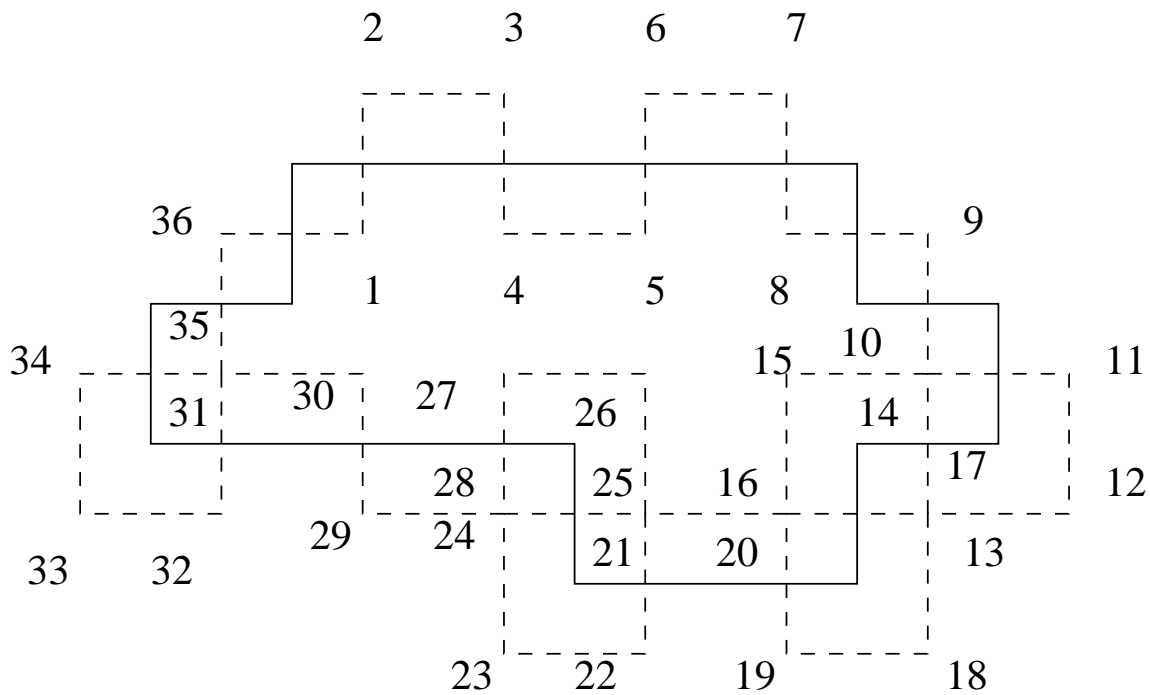


Figura 8.3: Ejemplo de Seguimiento de vontorno.

Seguimiento del contorno

Como el nombre sugiere, los algoritmos de seguimiento de contorno trazan las fronteras ordenando los puntos de tipo borde sucesivos. Un algoritmo simple para trazar fronteras cerradas en imágenes binarias es el siguiente:

1. Comenzar dentro de la región A, (por ejemplo, el primer píxel que encontramos de la región cuando hacemos un rastero por filas),
2. Girar a la izquierda y pasar al píxel siguiente si estamos dentro de la región A, en caso contrario girar a la derecha y pasar al píxel siguiente,
3. Continuar hasta que se llegue al punto del que se partió.

La figura 8.3 muestra un ejemplo de como se sigue un contorno. Es interesante observar que algunos píxeles del contorno aparecen más de una vez. El contorno puede definirse como los puntos por donde se entra o sale de la región.

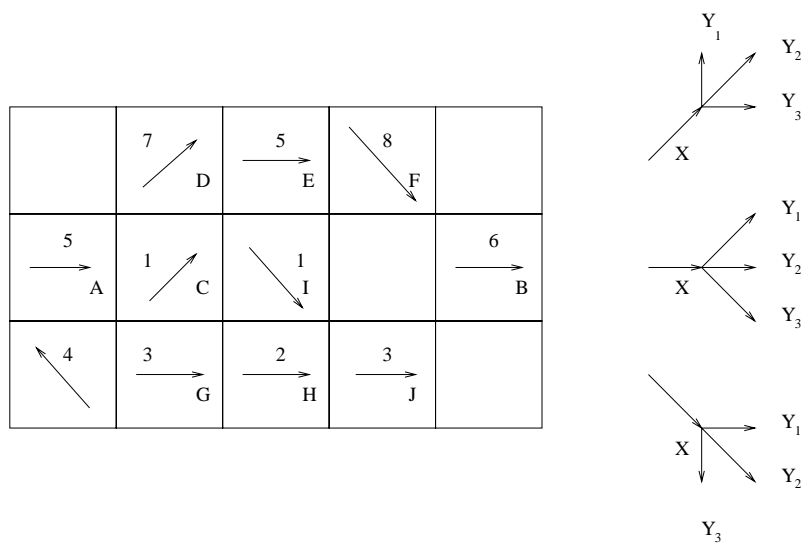


Figura 8.4: Unión heurística de bordes

Unión de bordes y búsqueda heurística en grafos

Una frontera puede ser también vista como un camino que se forma a través de un grafo construido uniendo los elementos bordes. Las reglas de unión dan la forma para conectar los bordes. Supongamos que un grafo con nodos en las posiciones $x_k, k = 1, 2, \dots$ se forma desde el nodo A al B , supongamos que tenemos una función de evaluación $\phi(x_k)$ que da el valor del camino de A a B cuando pasa por el nodo x_k . En algoritmos de búsqueda heurística, examinamos los sucesores del nodo de salida A y seleccionamos el que maximiza $\phi(\cdot)$. El nodo seleccionado se convierte ahora en el nodo de salida y se repite el proceso hasta que alcanzamos B . La sucesión de nodos seleccionada constituye la frontera. Observemos que el algoritmo no produce un óptimo global.

Veamos con un ejemplo estas ideas. Consideremos una matriz de tamaño 3×5 cuya magnitud de gradiente y dirección de contorno tangencial se muestran en la figura 8.4. Un píxel X se considera unido a Y si las correspondientes direcciones de contorno forman un ángulo menor de 90° (ver figura 8.4). Supongamos que la función de mérito que utilizamos $\phi(x)$ es la suma de los gradientes hasta llegar a x . Observemos que con este modelo, nuestro camino óptimo sería $ADEFB$, sin embargo el óptimo global es $ACDEFB$ (ver figura 8.4).

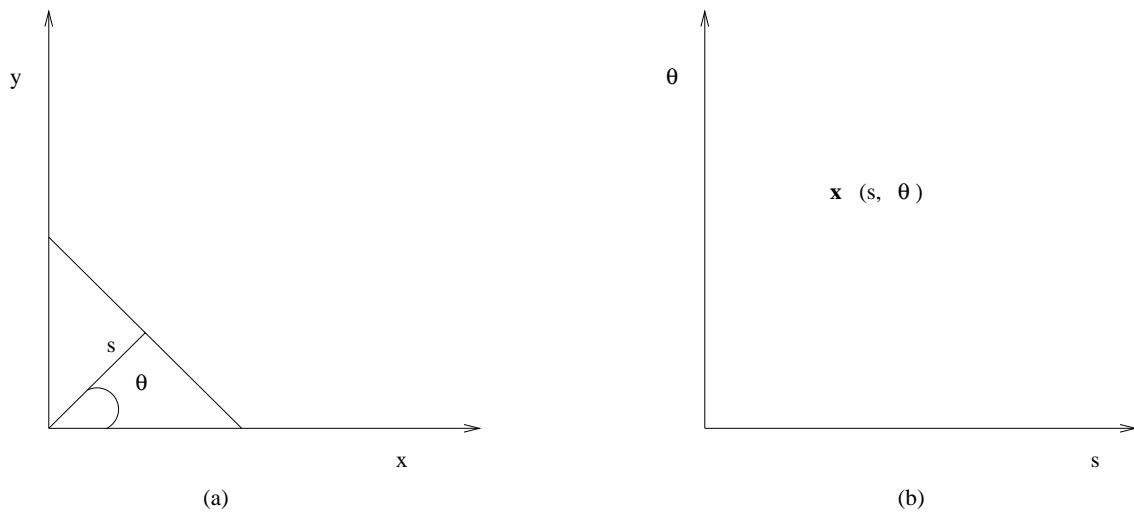


Figura 8.5: (a) Línea recta y (b) su transformada de Hough

8.4.2. Transformada de Hough

En esta sección consideraremos la unión de puntos examinando si están o no en una curva de una forma específica. Este método es obviamente una forma de considerar relaciones globales entre píxeles.

Supongamos que para n puntos de una imagen queremos encontrar subconjuntos de ellos que están sobre una línea recta. Una línea recta (ver figura 8.1a) a distancia s y con orientación θ puede representarse mediante

$$s = x \cos \theta + y \sin \theta \quad (8.1)$$

La transformada de Hough [19] de esta línea es un punto en el plano (s, θ) , esto es, todos los puntos de esta línea se aplican en un único punto (ver figura 8.1b). Este hecho puede usarse para detectar líneas rectas dado un conjunto de puntos frontera.

Supongamos que tenemos los puntos frontera $(x_i, y_i), i = 1, 2, \dots, n$. Para algunos valores cuantificados de los parámetros s y θ aplicaremos cada (x_i, y_i) en el espacio (s, θ) y contaremos $C(s, \theta)$, el número de puntos frontera que se aplican en una determinada localización (s, θ) , es decir, haremos

$$C(S_k, \theta_l) = C(S_k, \theta_l) + 1, \text{ si } s_k = x_i \cos \theta_l + y_i \sin \theta_l$$

Entonces los máximos locales de $C(s, \theta)$ proporcionan los diferentes segmentos de líneas rectas que pasan a través de un borde dado. Esta búsqueda bidimensional puede

reducirse a una búsqueda unidimensional si el gradiente θ_i en cada punto frontera se conoce. Teniendo en cuenta que derivando la ecuación de la recta dada en Eq. 8.5 se tiene que

$$\frac{dy}{dx} = -\cot\theta = \tan\left(\frac{\pi}{2} + \theta\right)$$

es obvio que sólo necesitamos evaluar el ángulo $\theta = \frac{\pi}{2} + \theta_i$.

Aunque el énfasis en esta subsección ha sido la aplicación de la transformada de Hough sobre líneas rectas, ésta es aplicable a funciones de la forma $g(\mathbf{v}, \mathbf{c}) = \mathbf{0}$ donde \mathbf{v} es un vector de coordenadas y \mathbf{c} es un vector de coeficientes. Por ejemplo, los puntos que están en la circunferencia

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2$$

pueden ser también detectados utilizando la aproximación anterior. El problema ahora es que tenemos más parámetros (tres en este caso) para localizar.

8.5. Segmentación Orientada a Regiones

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

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

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

8.5.1. Unión de Regiones

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

1. Definir una segmentación inicial que cumpla el criterio de homogeneidad,

0	0	5	6	7
1	1	5	8	7
0	<u>1</u>	5	<u>6</u>	7
2	0	7	6	6
0	1	5	6	5

(a)

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

(b)

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

(c)

Figura 8.6: Ejemplo de crecimiento de regiones a partir de puntos semilla.

2. Definir un criterio para unir regiones adyacentes,
3. Unir las regiones adyacentes si cumplen el criterio de unión. Parar cuando no puedan unirse dos regiones sin romper el criterio de homogeneidad.

La implementación más simple de este algoritmo empieza la unión de regiones comenzando la segmentación con regiones 2×2 , 4×4 u 8×8 . Las descripciones de las regiones se basan entonces en propiedades estadísticas de los niveles de gris, como por ejemplo en el histograma, la descripción de una región se compara con la de otra adyacente, si coinciden se unen las regiones y se computa el nuevo descriptor de la región. En caso contrario las regiones se marcan como que no se pueden unir. El proceso continua.

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

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

8.5.2. Unión de regiones por agregación de píxeles

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

Para ilustrar ese procedimiento consideremos la figura 8.6, donde los números dentro de las celdillas representan niveles de gris. Sean los puntos de coordenadas (3,2) y (3,4) las semillas que consideraremos. Si utilizamos estos dos puntos semilla tendremos una segmentación que consiste en dos regiones a lo más: una región R_1 asociada con la semilla (3,2) y otra R_2 asociada con (3,4). La propiedad P que utilizaremos para incluir un pixel en una región es que la diferencia en valor absoluto entre los niveles de gris del pixel y la semilla sea menor que un umbral T . Cualquier pixel que cumpla la propiedad dada con relación a los dos píxeles es asignado arbitrariamente a una de las regiones. La figura 8.6b muestra el resultado para $T = 3$. Las dos regiones segmentadas se notan a y b . Con $T = 8$ tendríamos una única segmentación como se muestra en la figura 8.6c.

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

8.5.3. Separación y Unión de Regiones

El procedimiento que acabamos de describir hace crecer las regiones a partir de un conjunto de puntos semilla. Una alternativa es subdividir una imagen inicialmente en un conjunto de regiones disjuntas y luego unirlos o separarlos en un intento de que todas ellas:

1. cubran la imagen completa,
2. cada una de ellas esté conectada,
3. sus intersecciones dos a dos sea el conjunto vacío,
4. que algún predicado de homogeneidad sea verdadero en cada región y
5. que la unión de dos cualesquiera de ellas no cumpla algún predicado de homogeneidad.

Sea R la imagen entera y seleccionemos un predicado de homogeneidad P . Supuesto que tenemos una imagen cuadrada una aproximación para segmentar R es subdividir la imagen sucesivamente en cuadrantes menores, de forma que tendremos que para cada región R_i , $P(R_i) = V$. Esto es, si $P(R) = F$, dividiremos el cuadrante en subcuadrantes

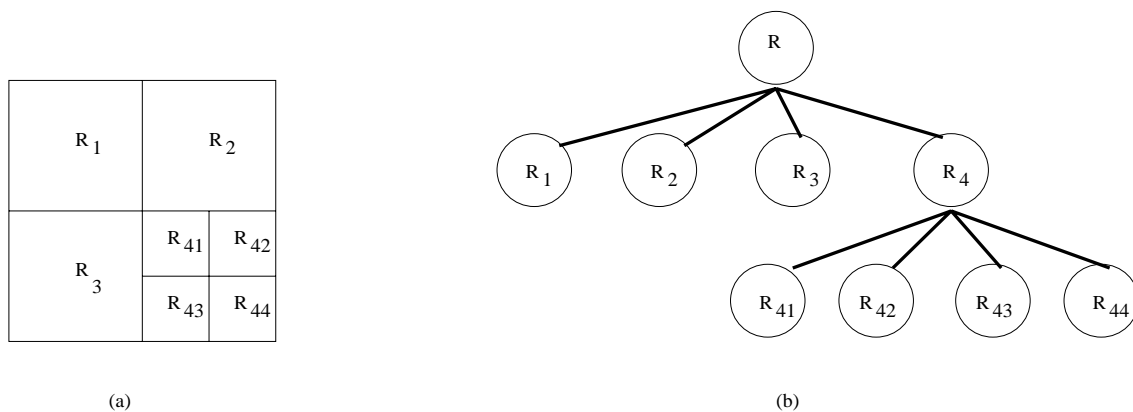


Figura 8.7: Ejemplo de quad-tree

y repetimos el proceso con los subcuadrantes obtenidos. Esta técnica de división puede representarse muy bien a través de los llamados quad-trees, esto es, un árbol en el que cada nodo tiene cuatro descendientes (ver figura 8.7). Observemos que la raíz del del árbol corresponde a la imagen original y que cada nodo es una subdivisión. En este caso, sólo R_4 se dividió más.

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

La discusión anterior puede resumirse en el procedimiento siguiente

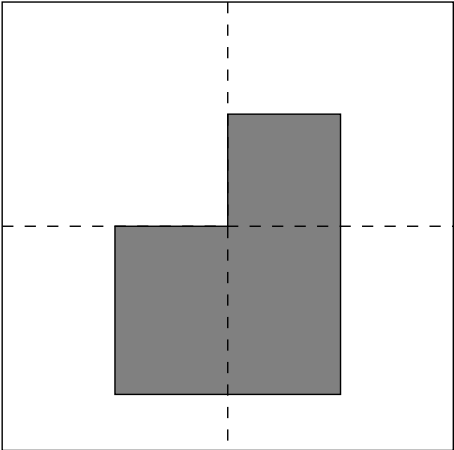
1. Dividir en cuatro regiones disjuntas cualquier región R_i para la que $P(R_i) = F$,
2. Unir cualesquiera dos regiones adyacentes R_j y R_k para las que $P(R_j \cup R_k) = V$,
3. Parar cuando no sea posible realizar mas uniones o divisiones.

La figura 8.8 ilustra el algoritmo de división y unión. La imagen consiste en un objeto único con fondo, con nivel de gris constante en el fondo como en el objeto. El predicado utilizado es $P(R_i) = V$ si todos los píxeles en la región R_i tienen la misma intensidad. Como $P(R) = F$ para la imagen completa, la dividimos como muestra la figura 8.8a. En el paso siguiente, sólo la región en la zona superior izquierda cumple el predicado y por tanto no es cambiada. Las otras tres sí lo son, ver figura 8.8b. En este momento podemos

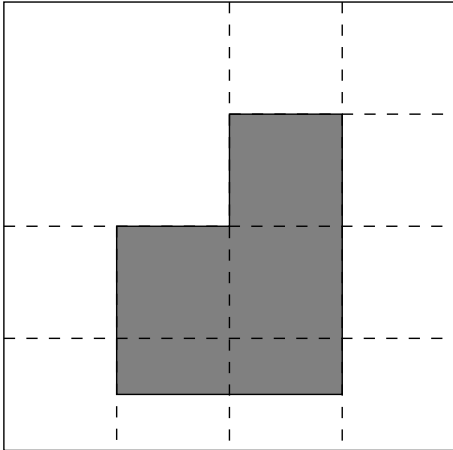
realizar un proceso de unión que no puede llevarse a cabo con los dos subcuadrantes que contienen la parte baja del objeto. La figura 8.8c muestra el resultado de la operación de partición y unión. En este punto todas las regiones satisfacen P y la unión de las regiones obtenidas por la última división produce el resultado final que se muestra en la figura 8.8(d).

8.6. Ejercicios

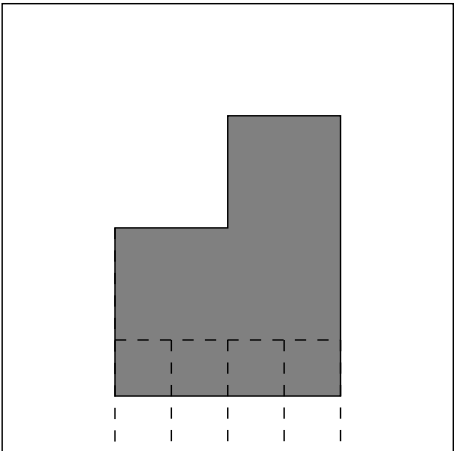
1. Implementar la técnica de umbralización para segmentación de regiones.
2. Implementar una técnica de extracción de bordes de una región a partir de la región.
3. Implementar un proceso de segmentación de regiones a partir de píxeles semilla.
4. Opcional: Implementar el método de unión y separación iterativa para la segmentación de regiones.



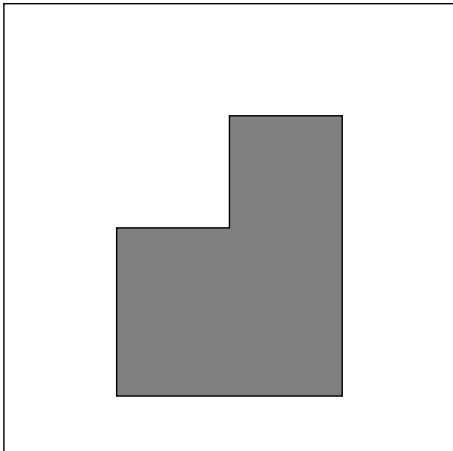
(a)



(b)



(c)



(d)

Figura 8.8: Proceso de separación y unión usando quad-trees.

Capítulo 9

Transformada de Hough

9.1. Motivación.

Si disponemos de la forma de las regiones o de los objetos de una imagen, la segmentación se puede considerar como el problema de localizar dichos objetos. Ejemplos clásicos son la búsqueda en aplicaciones industriales de objetos circulares, o la búsqueda de objetos de formas específicas en imágenes de satélite.

La clave del éxito en estos métodos es la unión de la información de los puntos de discontinuidad de la imagen con el conocimiento a priori que se tiene sobre la forma a localizar.

El hablar de la localización de objetos puede indicar que nuestro objetivo será la localización de una curva cerrada que delimita la frontera de una región en la imagen. Sin embargo, nuestro objetivo será más general ya que intentaremos extraer aquellas curvas, de forma conocida, que determinan la separación entre objetos.

Un método muy eficaz para resolver dicho problema es la *Transformada de Hough*, que se comporta con una gran robustez incluso sobre imágenes con graves problemas de solapamiento y ruido.

9.1.1. Problema.

El problema que queremos resolver consiste en diseñar un algoritmo para la localización de curvas en una imagen. Para ello, disponemos de:

- Una imagen I de niveles de gris y de dimensiones $N \times N$. A partir de ella, podemos

aplicar un algoritmo de extracción de discontinuidades dando lugar a un conjunto de m puntos frontera

$$\mathcal{P}_I = \{(x, y) / (x, y) \text{ frontera en } I\} \quad (9.1)$$

Este conjunto de puntos, que será el que usemos como entrada al algoritmo, nos informa sobre los lugares más probables por donde puede pasar una curva que separa dos objetos o regiones de la imagen. Así, nosotros los denominaremos el conjunto de *evidencias*.

- Un modelo de la curva o familia de curvas que queremos localizar. Este conjunto de curvas vendrá descrito en forma de una expresión analítica. La expresión general de una expresión analítica podemos escribirla como

$$f(x, v) = 0 \quad (9.2)$$

donde $v = (a_1, \dots, a_n)$ es el vector de parámetros de la curva, que corresponde a un punto del *espacio de parámetros*.

Como ejemplos, podemos considerar la rectas, círculos, elipses, etc. La extracción de este tipo de curvas será de gran utilidad para ciertas aplicaciones en las que la localización de estas *características* en la imagen puede servir como entrada para etapas posteriores del procesamiento¹.

Obviamente, para mucha aplicaciones, las curvas que determinan los bordes de objetos no pueden expresarse (al menos de una forma simple) de forma analítica. A pesar de ello, será posible obtener una generalización de la *Transformada de Hough* para estos casos.

9.1.2. Soluciones simples.

Algunas soluciones simples y a la vez de difícil aplicación en la práctica debido especialmente a problemas de eficiencia son por ejemplo:

¹Estas etapas posteriores podrían incluso corresponder a la aplicación de la *transformada de Hough* usando características de más alto nivel semántico que los puntos de frontera (véase Suchendra [3])

1. Supongamos que queremos localizar las rectas que aparecen en la imagen y que el número de puntos frontera obtenidos es n , es decir, $Cardinal(\mathcal{P}_I) = n$. Una posible solución es determinar una recta por cada par de puntos obtenidos y calcular los puntos que caen “cerca” de ella. Este algoritmo implicaría el cálculo de $\frac{n(n-1)}{2}$ líneas y para cada una de ellas del orden de n comprobaciones (para cada uno de los puntos restantes). En definitiva resultaría un algoritmo de eficiencia $O(n^3)$, la cual resulta demasiado alta para la mayoría de las aplicaciones.
2. Supongamos que queremos localizar un círculo de radio r en la imagen. Podemos construir una máscara de tamaño $2r \times 2r$ que contenga un 1 en cada uno de los puntos que corresponden al círculo y un 0 en el resto. Una posible solución es “pasear” la máscara por la imagen y calcular la correlación. Los valores más altos en el resultado indicarán posibles localizaciones del círculo. Desafortunadamente, este algoritmo sólo es posible aplicarlo para una determinada máscara (una sólo curva). El plantearlo para una familia de curvas implicaría realizar una operación de correlación para cada uno de los elementos de \mathcal{C} , lo que conllevaría a un algoritmo demasiado ineficiente.

Por supuesto, estos dos algoritmos simples, se han particularizado para los casos de las rectas y los círculos, aunque pueden ser generalizados para otras curvas, dando lugar a resultados igualmente poco satisfactorios.

9.1.3. Objetivo.

Como se puede observar, en las soluciones simples que acabamos de exponer, nuestra intención ha sido la de evaluar qué número de puntos de la imagen frontera, evidencian un determinado punto del espacio de parámetros. Como resultado del algoritmo, daremos el punto de este espacio con un mayor valor.

Nuestro objetivo será encontrar un método que a partir de la información de puntos frontera, obtenga un valor de acumulación de evidencias (votos) para cada posible tupla de valores del espacio de parámetros. La solución que buscamos vendrá dada por el máximo obtenido.

En la figura 9.1 podemos ver de forma gráfica esta idea, considerando los casos concretos de rectas y círculos. En la parte izquierda aparecen dos imágenes, que evidencian

la existencia de una recta y un círculo, respectivamente. En la parte de la derecha se representan sus respectivos espacios de parámetros, m - c para rectas y (a,b,r) para círculos.

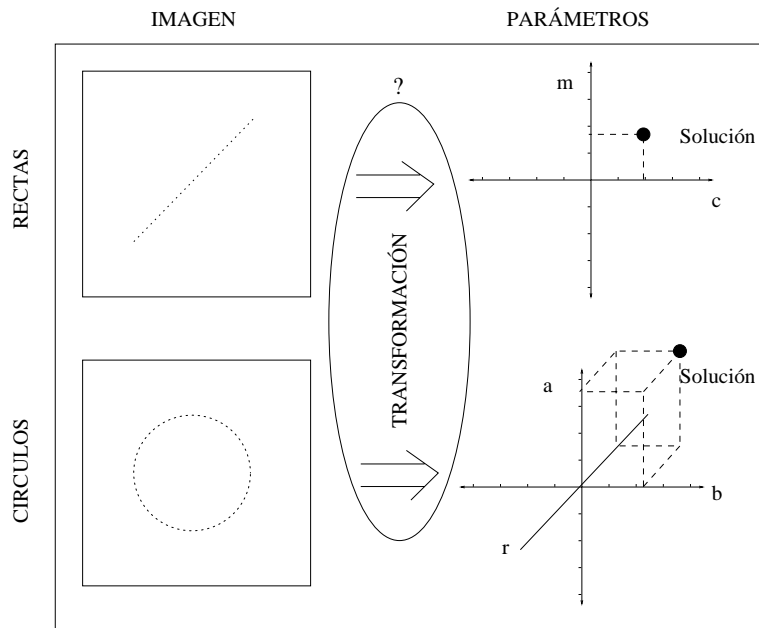


Figura 9.1: Motivación de la transformada de Hough.

Nuestro objetivo es determinar un algoritmo que en un tiempo razonable nos permita transformar, la información sobre localización de fronteras, al espacio de votaciones sobre los parámetros.

A continuación presentamos una solución a este problema, la *transformada de Hough*. La idea intuitiva que podemos tener en cuenta y que es base de la transformada de Hough es:

Recorreremos todos los puntos frontera, y para cada uno de ellos, acumularemos un voto en aquellas posiciones del espacio de parámetros que correspondan a una curva que pasa por dicho punto frontera.

9.2. Transformada de Hough.

Como hemos visto en los ejemplos simples del apartado anterior, la localización directa en el *espacio de la imagen* del conjunto de puntos que determina la localización de la curva en dicha imagen no es simple. Este problema se puede reformular a través de la

Transformada de Hough, la cual establece un método por el que se transforma el conjunto de evidencias que aparece en el *espacio de la imagen* al *espacio de los parámetros*. De esta manera, el problema de localizar un conjunto de puntos dispersos en el primer espacio se traduce en determinar un único punto en el segundo. Veamos a continuación cómo se puede llevar a cabo dicha transformación para distintas familias de curvas.

9.2.1. Detección de líneas.

Las fronteras en forma de líneas rectas son muy comunes en muchas aplicaciones de visión artificial. Considérese por ejemplo la gran cantidad de componentes industriales con líneas rectas, o la gran cantidad de líneas que aparecen en nuestro entorno (edificios, habitaciones, muebles, etc.). Por este motivo, la detección de este tipo de fronteras ha sido de especial importancia en muchos problemas.

La robustez y eficacia de la transformada de Hough la ha convertido prácticamente en el método más utilizado para la detección de estas líneas rectas desde que Hough, en 1962, patentara el dispositivo electrónico para detectar los caminos que seguían las partículas de alta energía (Hough[42]).

Ecuación explícita: espacio m-c.

La formulación inicial del método que se propone se realizó sobre el espacio (m, c) que corresponde a la pendiente y el desplazamiento de la recta

$$y = mx + c \tag{9.3}$$

Como se ha indicado anteriormente, nuestro objetivo será realizar una transformación de las evidencias del espacio de la imagen (espacio x-y) al espacio de las rectas (espacio m-c). El concepto básico que se utiliza para la detección de líneas en la transformada de Hough es la dualidad punto-línea.

Por supuesto, un sólo punto en el espacio m-c corresponde, como sabemos, a una recta en el espacio x-y. Por otro lado, un punto en el espacio x-y se puede definir con el punto donde cortan las infinitas líneas (para todos los valores de m) que pasan a través de él. Para un punto (x_o, y_o) , las rectas que pasan a través de él son las rectas que cumplen $y_o = mx_o + c$, es decir, los puntos (m, c) que corresponden a la recta $c = -mx_o + y_o$ del espacio m-c.

Si consideramos un conjunto de n puntos alineados \mathcal{P}_I , sólo existe una recta $y = m_0x + c_0$ que pasa a través de todos ellos. Por lo tanto, si dibujamos las n rectas del espacio m - c que corresponden a cada uno de los puntos obtendremos que todas ellas tienen en común el punto (m_0, c_0) .

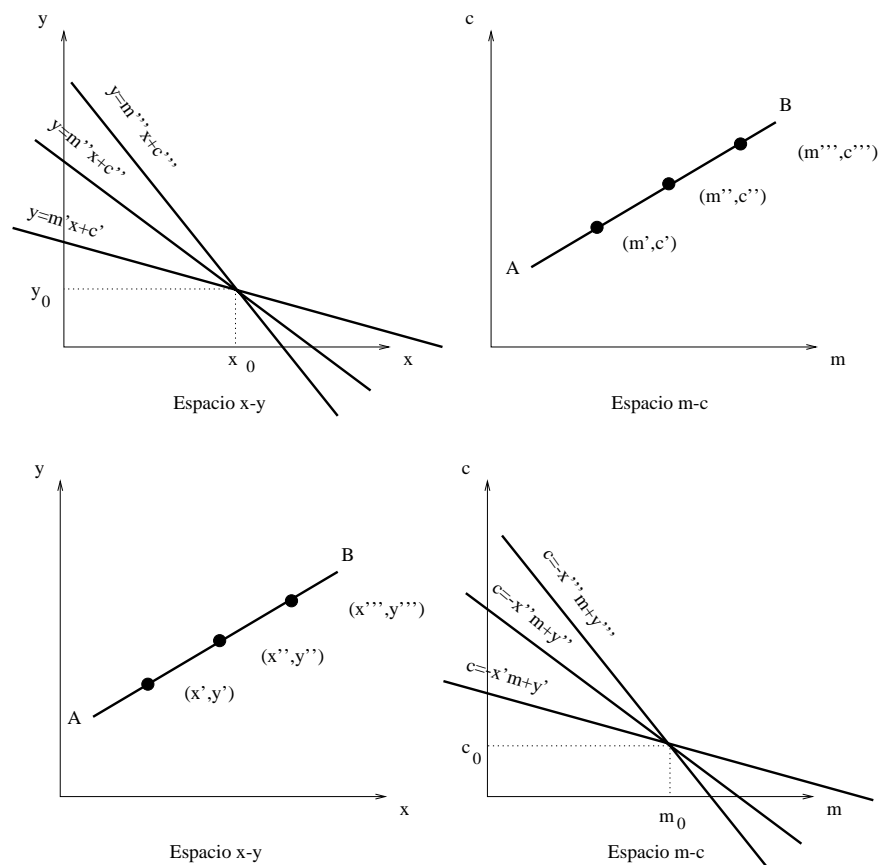


Figura 9.2: Dualidad punto-línea.

En la figura 9.2 podemos observar gráficamente esta dualidad. Vemos como el haz de líneas que pasa en un punto del espacio x - y corresponde a una línea del espacio m - c . De igual forma, vemos en la parte inferior cómo tres puntos alineados del espacio x - y corresponden a tres líneas que se cortan en un punto común, es decir, los puntos en una misma línea $y = m_0x + c_0$ corresponden a un haz de líneas en el espacio m - c sobre el punto (m_0, c_0) .

Esta situación nos sugiere un método para detectar líneas a partir de un conjunto de puntos \mathcal{P}_I . Para ello, interpretamos el espacio m - c como un espacio de votación, es decir, en el que para cada punto se acumulará un valor entero correspondiente al número de

votos que indican la presencia de esa recta en la imagen. Para calcular el número de votos en cada posición, hacemos que para cada punto de \mathcal{P}_I se incremente en uno todos los puntos que corresponden a la recta del espacio m-c (parte inferior de la figura 9.2).

Como resultado de este algoritmo, en cada punto (m, c) obtenemos un número de votos igual al número de puntos situados sobre la recta $y = mx + c$. Es decir, la búsqueda de un conjunto de puntos alineados en el espacio x-y se convierte en la búsqueda de un punto con un alto valor de acumulación en el espacio m-c (parte superior de la figura 9.2).

Obviamente, el algoritmo se aplica a un conjunto de puntos \mathcal{P}_I que corresponden a píxeles en la imagen I . Este algoritmo transforma estos puntos al espacio de parámetros A que resulta de una *discretización* del espacio continuo m-c. Para llevar a cabo esta discretización, será necesario establecer:

- Un rango de valores válidos para cada uno de los parámetros del espacio. En el caso de las rectas tendremos que determinar los intervalos

$$[m_{min}, m_{max}] \quad y \quad [c_{min}, c_{max}] \quad (9.4)$$

- Un incremento o precisión en cada parámetro. En el caso de las rectas tendremos que determinar los valores Δm y Δc

Estos valores dan lugar a un espacio de parámetros A de dimensiones

$$n_m \times n_c = \left(\frac{m_{max} - m_{min}}{\Delta m} + 1 \right) \times \left(\frac{c_{max} - c_{min}}{\Delta c} + 1 \right) \quad (9.5)$$

tal como muestra la figura 9.3, de forma que la posición $A[m, c]$ contiene el número de votos de la recta con parámetros (m, c) .

El algoritmo sería:

Algoritmo de detección de rectas (espacio m-c)

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en I .
2. Inicializar el acumulador $A[m, c]$ a cero.
3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para $m = m_{min}$ hasta $m = m_{max}$ hacer

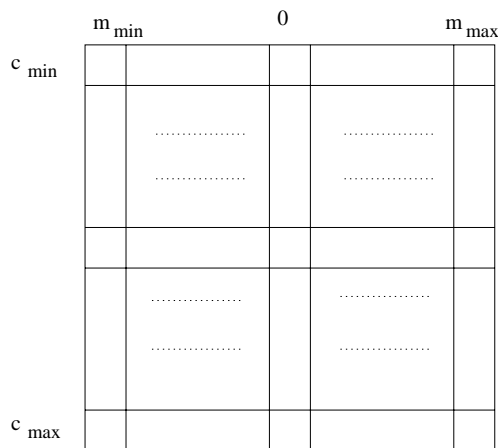


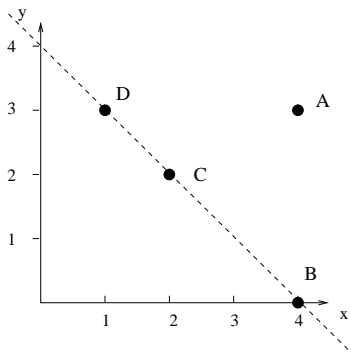
Figura 9.3: Discretización del espacio m-c.

- 1) $c = -xm + y$ (redondeando al valor más cercano de la discretización)
 - 2) $A[m, c] = A[m, c] + 1$
4. Los máximos del acumulador $A[m, c]$ corresponden a los puntos frontera que se encuentran alineados en la imagen I .

Un ejemplo de la aplicación de este método se presenta en la figura 9.4. En esta figura podemos ver:

- Espacio de la imagen y mejor solución. Aquí se representan los cuatro puntos (A, B, C, D) de entrada al algoritmo. Como se puede observar, la mejor solución es la recta que pasa por los puntos $\{B, C, D\}$.
- Puntos y rectas correspondientes. Considerando el haz de rectas que pasa por cada uno de los puntos, se puede representar la recta que corresponde a ese haz en el espacio de los parámetros. Aquí se muestran los puntos en el espacio de la imagen y su recta correspondiente en el de parámetros.
- Espacio de parámetros. Aquí mostramos la representación gráfica en el espacio continuo m-c de las cuatro rectas que se han obtenido. Se ve claramente que el punto $(-1, 4)$ se cruzan 3 rectas y por tanto estos valores corresponderán a la pendiente y desplazamiento de la recta de mejor solución.

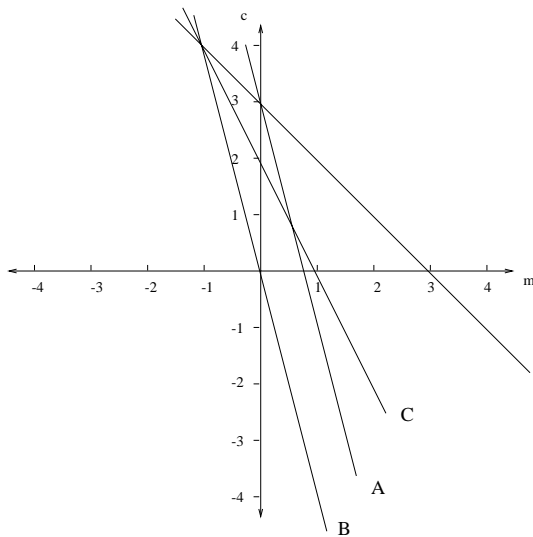
- Finalmente representamos una matriz que corresponde a la discretización del espacio m-c, es decir, al acumulador usado en el algoritmo. En este hemos tomado como valores de m los del rango $[-1, 3]$ con saltos de 0,5 y como valores de c los del rango $[0, 4]$ con saltos igualmente de valores 0,5. Se ha representado un círculo negro por cada voto. Como podemos ver, se han obtenido un máximo en la casilla $(-1, 4)$



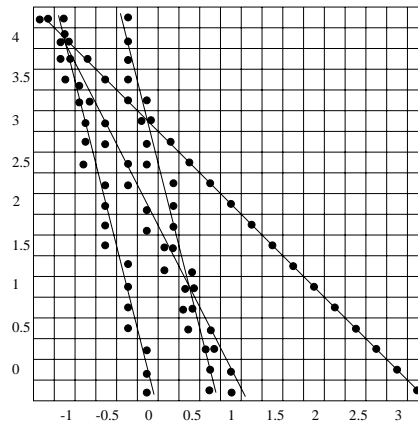
Espacio de la imagen y mejor solución

	Puntos		Rectas
	X	Y	$c=-xm+y$
A	4	3	$c=-4m+3$
B	4	0	$c=-4m+0$
C	2	2	$c=-2m+2$
D	1	3	$c=-1m+3$

Puntos y rectas correspondientes



Espacio de parámetros



Espacio de parámetros discretizado

Figura 9.4: Transformada de Hough usando el espacio m-c.

Fuentes de error en la Transformada de Hough.

Aunque desde un punto de vista teórico hemos podido observar la eficacia del método de la transformada de Hough, desde un punto de vista práctico, tendremos que tener

especial cuidado en su implementación debido a diferentes fuentes de error. Podemos distinguir las siguientes:

1. *Votación uniforme en el espacio de parámetros.* Consideremos el algoritmo expuesto en el apartado anterior y el ejemplo de la figura 9.4. En el algoritmo recorreremos el rango del parámetro m para obtener los valores correspondientes a c y realizar la votación. Una primera implementación de este método podría llevarnos a recorrer cada uno de los valores de m representado en el acumulador y para cada uno de ellos obtener el punto de votación. Si observamos el espacio de parámetros discretizado de la figura vemos que esto nos llevaría a un grave error ya que algunas de las rectas² discretizadas no podría representarse de forma apropiada. El resultado sería una recta “a saltos” pues para un valor concreto de m le corresponden varios de c . Esto puede provocar errores debido a que no se ha votado en algunos puntos válidos del espacio de parámetros y puede llevarnos a eliminar votos de la posición donde deberíamos obtener el máximo que buscamos.

La solución a este problema es una correcta representación de la curva en el espacio de parámetros. Esta se puede obtener aplicando algoritmos de dibujo de curvas en un espacio discreto, tales como los usados en informática gráfica. Así, el bucle interior del algoritmo presentado debería reescribirse de acuerdo a alguno de ellos. Por ejemplo, usar el algoritmo de Bresenham para obtener los puntos de la recta $c = -mx + y$ en el espacio discreto $A[m, c]$.

2. *Discretización del espacio de parámetros.* Como podemos observar en la figura 9.4, Existen tres puntos destacados en el espacio continuo m - c , correspondientes a un corte de 3 rectas y dos de 2 rectas. Sin embargo, en el espacio de parámetros discretizado existen más de tres. Este error se debe a la gruesa discretización que se ha llevado a cabo. Si los parámetros no se muestrean suficientemente, en algunas celdas podremos acumular votos que nos lleven erróneamente a la detección de máximos.

Podemos plantear una solución mediante el aumento de la precisión del espacio de parámetros a cambio de recursos en tiempo y espacio. Efectivamente, esta solución provocaría una mayor dispersión de votos en varias casillas evitando la acumulación excesiva que comentamos. Sin embargo, debemos tener en cuenta que

²Las rectas que generarían problemas serían aquellas cuya pendiente fuese mayor que 1 o menor que

- Existe un error de discretización en el espacio de parámetros. Consideremos por ejemplo, que los parámetros de la curva a localizar corresponden a un valor intermedio a dos celdas. Debido a la aproximación de las curvas teóricas cuando las dibujamos podemos obtener una distribución de votos en varias casillas distintas. Esta situación dificultaría la obtención de un máximo en ese punto. Nótese que esta dispersión de votos puede aumentarse si incrementamos la precisión del muestreo de los parámetros.
- Existe un error de discretización en el espacio de la imagen. La situación de la figura 9.4 es ideal en el sentido de que los tres puntos alineados están situados exactamente en la misma recta. Además, esta recta se representa justo en una celda del espacio de parámetros. En la práctica, el conjunto de puntos de la imagen se obtendrá probablemente tras la aplicación de un detector de fronteras que unido a los errores de discretización dará lugar a un conjunto de puntos lejos de ser ideal. Este ruido en los datos de entrada provoca igualmente una dispersión de los votos que se ve incrementada si aumentamos la precisión del muestreo del espacio de parámetros.

En definitiva, los errores de discretización nos llevan por un lado a una acumulación excesiva de votos y, por otro, a una dispersión de éstos.

En la práctica, la discretización del espacio de parámetros se realiza hasta un nivel de precisión suficientemente alto como para evitar la acumulación excesiva de votos y obtener una cota de error aceptable para la aplicación. La localización de los máximos puede verse afectada por la *dispersión de máximos*, por lo que dicha fase se acompaña de algún procedimiento de postprocesado del espacio de parámetros como un alisado. Finalmente, se obtienen como solución los puntos de valores altos que corresponden a máximos locales (dado un máximo, se eliminan como candidatos los puntos de su entorno).

3. *Expresión analítica de la curva a localizar.* Para una misma curva, podemos escoger entre varias expresiones analíticas para representarla. Obviamente, esta elección afecta directamente a la forma y significado del espacio de parámetros. Dos características deseables para dichos parámetros son que estén acotados y que tengan una precisión uniforme a lo largo de los distintos valores de rotación de la curva.

Si volvemos sobre el ejemplo de la figura 9.4, vemos que en el espacio de parámetros

continuo se observan otros dos cortes en los que sólo intervienen dos rectas. Estos corresponden a las rectas A, D y A, C . Es interesante destacar que no aparece la recta A, B ya que esta tendría que estar representada en un valor de pendiente infinita. Este es un ejemplo claro en el que el rango de valores de m no está acotado (la recta A, B no es representable) y la precisión no es uniforme.

Este problema sobre la representación que se ha usado nos lleva a realizar una nueva propuesta, en la que podremos representar fácilmente cualquier recta de la imagen en un espacio de parámetros acotado.

Ecuación “normal”: espacio $\theta - \rho$

Hemos podido ver la forma en que podríamos aplicar la transformada de Hough usando el espacio m-c. En este apartado vamos a plantear una nueva representación para obtener un espacio de parámetros más adecuado para la detección de rectas.

En primer lugar podríamos proponer una segunda representación basada en el espacio m-c que nos evitara el problema del rango no acotado. Esta solución consiste en establecer dos conjuntos de puntos, aquellos que tienen pendientes (en valor absoluto) menores que 1 y los que tienen mayores o iguales a 1. Para poder representar los valores altos en este segundo caso, la ecuación 9.3 se reemplaza por

$$x = \tilde{m} y + \tilde{c} \quad (9.6)$$

donde

$$\tilde{m} = \frac{1}{m} \quad (9.7)$$

y por tanto los valores de la pendiente vuelven a estar acotados en el rango $[-1, 0, 1, 0]$.

Sin embargo, no vamos a adoptar esta solución sino la propuesta por Duda y Hart [9] en la que se propone usar la ecuación “normal” definida sobre los parámetros (θ, ρ)

$$\rho = x \cos \theta + y \sin \theta \quad (9.8)$$

En la figura 9.5 se puede observar la interpretación de los parámetros de la recta y la discretización del plano $\theta - \rho$ en celdas. Como se puede observar, ρ es la distancia del origen a la recta (distancia del origen al punto más cercano) y θ corresponde al ángulo formado por la recta, que une el origen con el punto más cercano, y el eje de abscisas.

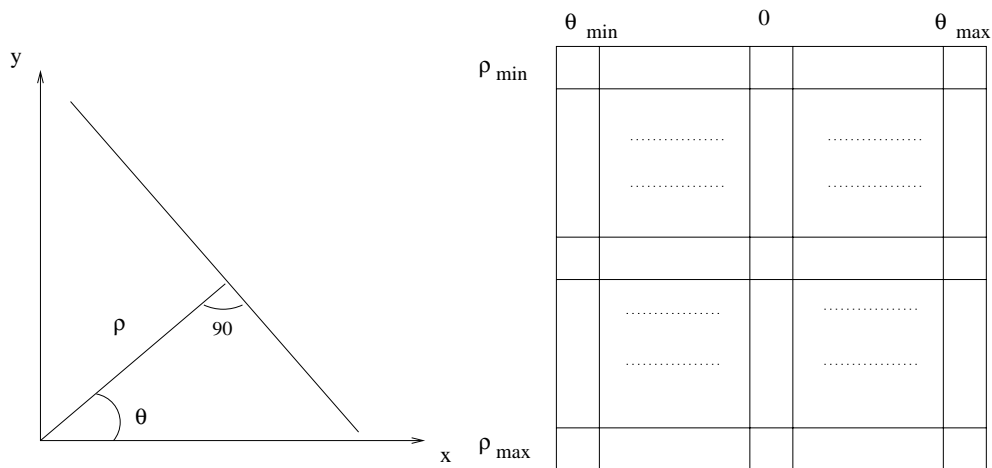


Figura 9.5: Representación “normal” de la recta y discretización del espacio θ, ρ .

En el caso anterior (espacio m-c), un haz de líneas en el espacio de la imagen se convertía en una línea en el espacio de los parámetros. En esta formulación, un haz de líneas se transforma en una curva senoidal.

$$(x_0, y_0) \Rightarrow \begin{cases} c = -mx_0 + y_0 & \text{Si usamos el espacio } m - c \\ \rho = x_0 \cos \theta + y_0 \sin \theta & \text{Si usamos el espacio } \theta - \rho \end{cases} \quad (9.9)$$

En la figura 9.6 se muestra un ejemplo de la transformada de Hough para localizar las rectas usando el espacio $\theta - \rho$. En la parte superior izquierda se han representado los cinco puntos de entrada al algoritmo. En la parte superior derecha se han pintado con trazo discontinuo las dos rectas principales, es decir, las que acumulan más votos (en este ejemplo simple, sólo tres votos). En la parte inferior se muestra el espacio de parámetros.

Como se puede observar, aparecen dos puntos destacados (marcados con dos círculos y las letras A, B), correspondientes a las rectas A y B que se pretendían buscar. En cada uno de estos puntos se cruzan tres curvas que provienen de cada uno de los puntos que pertenecen a cada recta. Las dos curvas principales por tanto son las correspondientes a parámetros $(179, \frac{\pi}{4})$, es decir,

$$179 = \sqrt{127^2 + 127^2} = x \cos \frac{\pi}{4} + y \sin \frac{\pi}{4} \quad (9.10)$$

y la de parámetros $(0, \frac{3\pi}{4})$, es decir,

$$0 = x \cos \frac{3\pi}{4} + y \sin \frac{3\pi}{4} \quad (9.11)$$

Por otro lado hemos marcado con círculos de menor radio otros cruces de las curvas, en este caso de sólo dos de ellas. Estos puntos coinciden con las cuatro rectas de dos votos que pueden obtenerse de los datos de entrada (dos verticales y dos horizontales). Es de destacar que los puntos marcados en $(0,0)$ y $(\pi,0)$ se refieren a la misma recta así como los puntos $(0,254)$ y $(\pi,-254)$. Nótese que este efecto se debe a la forma en que los parámetros ρ y θ cambian en los límites 0 y π .

9.2.2. Ejemplos.

Ejemplo 1: 4 líneas.

En primer lugar vamos a estudiar un ejemplo muy simple. Es el caso de una imagen en la que sólo aparecen 4 líneas rectas. El ejemplo se muestra en la figura 9.7. Como podemos ver, en la parte izquierda se muestra una imagen de tamaño 200×200 con cuatro líneas numeradas. En la parte derecha se muestra el espacio de parámetros correspondiente, donde se ha señalado la localización de los 4 máximos de cada una de las rectas de entrada. También se puede observar un posible máximo (no señalado) en coordenadas $(\pi, -100)$ que corresponde a la recta número 1 y que es consecuencia de la “continuidad reflejada” del espacio de parámetros (corresponde al máximo $(0,100)$).

Ejemplo 2: imagen de una casa.

El segundo ejemplo (figura 9.8) corresponde a la imagen de una casa. En este caso, se han localizado las fronteras de la imagen original, sobre las que se ha aplicado el algoritmo de detección de líneas con la condición de que aparezca cualquier líneas que acumule más de 10 puntos. Las imágenes que se muestran son:

- A.- Imagen original de dimensiones 256×256 con 256 niveles de grís.
- B.- Fronteras obtenidas desde la imagen A.
- C.- Espacio de parámetros resultante de la imagen B.
- D.- Líneas detectadas.
- E.- Segmentos de línea originales detectados sobre la imagen D.
- F.- Segmentos de línea determinados por la existencia de puntos frontera de la imagen B.

Ejemplo 3: imagen ruidosa.

En este último ejemplo (figura 9.9), se muestra el resultado a partir de una imagen que contiene 6 líneas y que ha sido contaminada por un 10 % de ruido. Las imágenes que se muestran son:

- A.- Imagen original de 6 líneas con ruido.
- B.- Espacio de parámetros obtenido desde la imagen A.
- C.- Seis primeras líneas que se han obtenido desde el espacio de parámetros B.
- D.- Recuperación de segmentos desde las imágenes A y C. El algoritmo aplicado sigue la línea obtenida y dibuja todos los segmentos de línea continuos que tengan una longitud mínima.

Como se ha indicado, la condición de extracción de máximos para este caso ha sido que hemos establecido a priori que queríamos obtener 6 líneas. Obviamente, en una aplicación real, es difícil establecer una condición general que sea válida para todas las entradas. Nótese que en la imagen ruidosa pueden existir un gran número de rectas que, conteniendo un número suficiente de puntos, sólo estén compuestas por ruido y por tanto no sean válidas al no tener continuidad.

En la práctica se puede establecer un número mínimo de puntos para obtener una respuesta positiva en la detección de la línea, y posteriormente aplicar algún algoritmo de extracción de los puntos que componen el segmento en la imagen de fronteras para confirmar su existencia (los resultados de esta extracción se muestran en la figura 9.8-F y figura 9.9-D).

9.2.3. Generalización a una curva cualquiera.

Es sencillo llevar a cabo una generalización del algoritmo que se ha presentado para curvas generales definidas por una expresión analítica de la forma

$$f(x, y, v) = 0 \quad (9.12)$$

donde $v = (v_1, \dots, v_p)$ es el vector de parámetros de la curva y (x, y) es un punto en el espacio de la imagen.

Al igual que en el caso de la detección de líneas, tendremos que definir un acumulador A que resulte de la discretización del espacio continuo de los parámetros. Para ello será necesario

- Discretizar el espacio p -dimensional de parámetros de la curva, determinando valores mínimos, máximos e incrementos para cada parámetro V_i

$$v_{i,min}, v_{i,max}, \Delta v_i \quad (9.13)$$

dando lugar a un conjunto discreto de n_{V_i} valores para el parámetro i -ésimo.

- Definir el espacio discreto p -dimensional $A[V_1, \dots, V_p]$ de dimensiones $n_{V_1} \times \dots \times n_{V_p}$.

Además, supongamos una ecuación $V_p = F(V_1, \dots, V_{p-1}, x, y)$ que relaciona el punto frontera (x, y) con la curva de parámetros (V_1, \dots, V_p) a partir de la ecuación (9.12).

Así, la generalización del algoritmo presentado en la página 185 sería:

Algoritmo de detección de curvas con expresión analítica

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
 2. Se inicializa el acumulador A a cero.
 3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para toda tupla (V_1, \dots, V_{p-1}) del acumulador hacer
 - 1) $V_p = F(V_1, \dots, V_{p-1}, x, y)$
 - 2) $A[V_1, \dots, V_p] = A[V_1, \dots, V_p] + 1$
 4. Los máximos del acumulador $A[V_1, \dots, V_p]$ corresponden a las curvas en la imagen I .
-

Sin embargo, en la práctica puede ser más interesante usar una representación paramétrica de la curva que facilite el cálculo de las posiciones a incrementar en el acumulador. Además, estas representaciones pueden facilitar el uso de algoritmos de informática gráfica para asegurar la votación uniforme y mejorar la eficiencia (véase la sección 9.2.1 en la página 188).

Para ello, se definen dos funciones introduciendo una variable independiente t

$$V_p = f_1(V_1, \dots, V_{p-2}, x, y, t) \quad (9.14)$$

$$V_{p-1} = f_2(V_1, \dots, V_{p-2}, x, y, t) \quad (9.15)$$

para valores de t en el rango $[0, 2\pi]$.

Una vez que se han fijado los valores de $(V_1, \dots, V_{p-2}, x, y)$, (V_p, V_{p-1}) es un plano 2-dimensional y las ecuaciones (9.14) y (9.15) definen una curva que se puede dibujar en dicho plano usando algoritmos de informática gráfica (optimizados por los investigadores en este campo).

Es interesante destacar la elección de los parámetros (V_p, V_{p-1}) como el desplazamiento de la curva en el plano x-y. En este caso, la precisión en estos parámetros podemos relacionarla directamente con la de la imagen y podemos establecer $(\Delta V_p, \Delta V_{p-1}) = (\Delta x, \Delta y)$.

Si renombramos estos dos parámetros como (a, b) y reformulamos las ecuaciones de la curva como

$$(x, y) = (a, b) + (f_x(V_1, \dots, V_{p-2}, t), f_y(V_1, \dots, V_{p-2}, t)) \quad 0 \leq t \leq 2\pi \quad (9.16)$$

podemos proponer el siguiente algoritmo

Algoritmo de detección de curvas (paramétrica)

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
2. Se inicializa el acumulador A a cero.
3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para toda tupla (V_1, \dots, V_{p-2}) del acumulador hacer
 - 1) Para todo t tal que $0 \leq t \leq 2\pi$ hacer

$$a' \quad a = \text{Cuantizar}(x - f_x(V_1, \dots, V_{p-2}, t))$$

$$b' \quad b = \text{Cuantizar}(y - f_y(V_1, \dots, V_{p-2}, t))$$

$$c' \quad A[V_1, \dots, V_{p-2}, a, b] = A[V_1, \dots, V_{p-2}, a, b] + 1$$
4. Los máximos del acumulador $A[V_1, \dots, V_p]$ corresponden a las curvas en la imagen I .

Si estudiamos detenidamente los pasos que realiza el bucle más interno, nos podemos dar cuenta de que no es más que el dibujo de la curva

$$(-f_x(V_1, \dots, V_{p-2}, t), -f_y(V_1, \dots, V_{p-2}, t)) \quad 0 \leq t \leq 2\pi \quad (9.17)$$

centrada en el punto (x, y) , en el plano $A[V_1, \dots, V_{p-2}]$.

Detección de círculos.

En este apartado vamos a ver como la transformada de Hough para líneas puede formularse para la detección de círculos en una imagen siguiendo la generalización que acabamos de exponer.

La ecuación no paramétrica de la curva a localizar se puede escribir como

$$f(x, y, a, b, r) = (x - a)^2 + (y - b)^2 - r^2 = 0 \quad (9.18)$$

donde (a, b) determina el centro del círculo (el rango de valores de este punto central puede ser cualquier valor (x, y) de la imagen) y r es el radio del círculo (en un rango $[1, n_r]$).

Como vemos, el espacio de parámetros es ahora 3-dimensional y por tanto será necesario definir un acumulador de la forma $A[a, b, r]$ de dimensiones $N \times N \times n_r$.

Esta ecuación puede reescribirse para obtener un parámetro en función de los demás de la siguiente forma

$$r = +\sqrt{(x - a)^2 + (y - b)^2} \quad (9.19)$$

Por lo tanto, el algoritmo queda definitivamente como

Algoritmo de detección de círculos (no paramétrica)

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
2. Para todo (a, b, r) tal que $a \in \{0, \dots, N - 1\}$, $b \in \{0, \dots, N - 1\}$, $r \in \{1, \dots, n_r - 1\}$ hacer
 - a) $A[a, b, r] = 0$

3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para todo (a, b) tal que $a \in \{0, \dots, N-1\}, b \in \{0, \dots, N-1\}$ hacer
 - 1) $r = \text{Cuantizar}(\sqrt{(x-a)^2 + (y-b)^2})$
 - 2) $A[a, b, r] = A[a, b, r] + 1$
 4. Los máximos del acumulador $A[a, b, r]$ corresponden a las curvas en la imagen I .
-

Sin embargo, también podemos usar las ecuaciones paramétricas de la curva. En este caso, el círculo puede expresarse como

$$a = x - r \cos t \quad (9.20)$$

$$b = y - r \sin t \quad (9.21)$$

para valores de t en el rango $[0, 2\pi]$.

Por tanto podemos construir un algoritmo usando estas expresiones. Éste se podría escribir como

Algoritmo de detección de círculos (paramétrica)

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
 2. Para todo (a, b, r) tal que $a \in \{0, \dots, N-1\}, b \in \{0, \dots, N-1\}, r \in \{1, \dots, n_r-1\}$ hacer
 - a) $A[a, b, r] = 0$
 3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para todo (r, t) tal que $r \in \{1, \dots, n_r-1\}, t \in \{0, \dots, 2\pi\}$ hacer
 - 1) $a = \text{Cuantizar}(x - r \cos t)$
 - 2) $b = \text{Cuantizar}(y - r \sin t)$
 - 3) $A[a, b, r] = A[a, b, r] + 1$
 4. Los máximos del acumulador $A[a, b, r]$ corresponden a las curvas en la imagen I .
-

Nótese que el bucle interno no es más que el dibujo de los puntos de una circunferencia con centro en (x, y) y radio r en el plano (a, b) . Por tanto podremos usar un algoritmo de dibujo rápido para localizar las celdas que tienen que modificarse.

Un ejemplo de la aplicación del algoritmo a una imagen con 6 puntos se muestra en la figura 9.10. En esta figura se muestran los 6 círculos a localizar, la circunferencia con mayor acumulación de votos (centro $(100,100)$ y radio 50), y finalmente el espacio de parámetros con valores de desplazamiento del centro en el rango $[0,200]$ y valores de radio $\{40, 50, 60\}$. Se puede observar como el máximo del espacio de parámetros se encuentra en la posición $(a, b, r) = (100, 100, 50)$.

Detección de elipses.

Para representar una elipse, podemos utilizar los parámetros (a, b, c, d, θ) , donde (a, b) indica el centro de la elipse, c es el semieje mayor, y d es el semieje menor. En la figura 9.11 se muestra gráficamente el significado de estos parámetros.

La ecuación paramétrica en forma matricial que podemos usar para construir el algoritmo es

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} c & 0 \\ 0 & d \end{pmatrix} \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \quad (9.22)$$

donde, de igual forma que en el caso del círculo, hemos escrito los dos parámetros de desplazamiento en función del resto y t toma valores en el rango $[0, 2\pi]$. Esta ecuación nos da lugar al siguiente algoritmo

Algoritmo de detección de elipses

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
2. Para todo (a, b, c, d, θ) del espacio paramétrico discretizado hacer
 - a) $A[a, b, c, d, \theta] = 0$
3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para todo (c, d, θ, t) hacer
 - 1) $a = \text{Cuantizar}(x - (c \cos \theta \cos t - d \sin \theta \sin t))$

$$2) \quad b = \text{Cuantizar}(y - (c \operatorname{sen} \theta \cos t + d \cos \theta \operatorname{sen} t))$$

$$3) \quad A[a, b, c, d, \theta] = A[a, b, c, d, \theta] + 1$$

4. Los máximos del acumulador $A[a, b, c, d, \theta]$ corresponden a las elipses en la imagen I .
-

Nótese de nuevo, que el bucle interior corresponde al dibujo de una elipse de centro (x, y) , semieje mayor c , semieje menor d y ángulo de inclinación θ . Por tanto, los cálculos que hay que realizar no son necesarios si sustituimos este bucle por un algoritmo de dibujo de elipses en el plano (a, b) para cada tupla de valores (c, d, θ) considerada.

Generalización a una curva dibujada.

Hasta ahora hemos estudiado la aplicación de la transformada de Hough para curvas con expresión analítica. En este apartado vamos a considerar el problema de buscar un objeto que no tiene una forma analítica simple, sino un determinado contorno³.

Consideremos una curva C en el plano. Podemos seleccionar un punto de referencia en el plano (a, b) (por ejemplo, el centroide) para definir la siguiente ecuación paramétrica de la curva:

$$(x, y) = (a, b) + (f_x(t), f_y(t)) \quad 0 \leq t \leq 2\pi \quad (9.23)$$

La aplicación del algoritmo a la expresión de la curva de la ecuación (9.23) es directa. Si definimos un acumulador $A[a, b]$ para todos los posibles desplazamientos con una precisión igual a la de la imagen, el algoritmo quedaría:

Algoritmo de desplazamiento de una curva cualquiera

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
2. Se inicializa el acumulador A a cero.
3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para todo t tal que $0 \leq t \leq 2\pi$ hacer

³Nótese que vamos a hablar de curvas cerradas por su relación con el contorno de un objeto aunque el algoritmo será válido igualmente para cualquier otra curva.

- 1) $a = \text{Cuantizar}(x - f_x(t))$
- 2) $b = \text{Cuantizar}(y - f_y(t))$
- 3) $A[a, b] = A[a, b] + 1$

4. Los máximos del acumulador $A[a, b]$ corresponden a las curvas en la imagen I .

Si estudiamos la operación que se lleva a cabo el bucle más interior, veremos que sólo se trata del dibujo de una curva centrada en el punto (x, y) sobre el plano A . Esta curva es $(-f_x(t), -f_y(t))$, exactamente la misma que la de la ecuación 9.23 reflejada con respecto a su punto central o de referencia, como era de esperar. Nótese que sólo hemos reescrito el algoritmo que habíamos propuesto anteriormente usando la ecuación (9.23). De hecho, para los casos de círculo y elipse que hemos estudiado, se comentó que el bucle interior correspondía al dibujo de la circunferencia o elipse correspondiente aunque sin referirnos a la necesidad de reflejarlo con respecto al punto central ya que el resultado era la misma curva.

Supongamos que disponemos de una curva C dibujada sobre una imagen. Es posible, por tanto, proponer el siguiente algoritmo:

 Algoritmo de desplazamiento de una curva dibujada

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
 2. Se inicializa el acumulador A a cero.
 3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Dibujar C reflejada sobre (x, y) en el plano A
 4. Los máximos del acumulador $A[a, b]$ corresponden a las curvas en la imagen I .
-

En la figura 9.12 se presenta un ejemplo de aplicación de este algoritmo. En la parte superior izquierda tenemos la curva a localizar, a la que se le ha añadido una cruz con línea discontinua indicando el punto de referencia que hemos escogido. En la parte superior derecha, se presentan los puntos que serán entrada al algoritmo junto con la curva sobre ellos para señalar la posición $(100, 100)$ como la mejor localización. En la parte inferior derecha se muestra la curva reflejada con respecto a su punto de referencia, es decir, la que dibujaremos centrada en cada uno de los puntos de evidencia. Finalmente, en la parte inferior derecha se muestra el espacio de parámetros resultante, en el que se puede ver como el conjunto de curvas coinciden en señalar el punto $(100, 100)$ como solución. En este punto se encontrará el valor máximo (11) y será la solución final, como era de esperar.

Introducción de escala y rotación.

En la práctica, la localización de una curva cualquiera no suele aparecer simplificada a la determinación de los parámetros de desplazamiento (a, b) sino que puede venir afectada por una transformación de escala y rotación.

La introducción de estos dos parámetros en los cálculos es trivial ya que sólo tenemos que considerar la forma en que se incluían un número variable de parámetros en las secciones anteriores. Definamos el espacio de parámetros $A[e, r, a, b]$ de parámetros escala, rotación desplazamiento en x y desplazamiento en y . Si disponemos de una curva C dibujada sobre una imagen, el algoritmo para su localización teniendo en cuenta escala y rotación es:

 Algoritmo de escala, rotación y desplazamiento de una curva

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
 2. Se inicializa el acumulador A a cero.
 3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para todo (e, r) hacer
 - 1) Dibujar C escalada con e , rotada r y reflejada sobre (x, y) en el plano $A[e, r]$
 4. Los máximos del acumulador $A[e, r, a, b]$ corresponden a las curvas en la imagen I .
-

Por supuesto, la introducción de nuevos parámetros de transformación de la curva se tendrá en cuenta introduciendo cambios similares.

9.2.4. Ejemplos.

En la figura 9.13 se incluyen tres ejemplos de la generalización de la transformada de Hough:

1. Ejemplo de detección de círculos: la imagen de la izquierda corresponde a las fronteras extraídas. A continuación se presenta el espacio de parámetros para un radio fijo y finalmente la solución obtenida sobre la imagen original.
2. Ejemplo de detección de una curva dibujada: Se presenta la forma dibujada a localizar (izquierda) y el contorno (punteado) sobre una imagen aérea en la que se ha localizado.
3. Una imagen de células, donde se aplica la transformada de Hough para la detección de elipses. La imagen original se presenta en la imagen izquierda mientras que las elipses correspondientes se presentan en la parte derecha.

9.3. Eficiencia de la Transformada de Hough

9.3.1. Correlación vs. Transformada de Hough

En la segunda de las soluciones que hemos visto en el primer apartado del tema (soluciones simples a la localización de curvas) proponíamos calcular la correlación de

cada una de las curvas con la imagen y escoger los puntos de máximo para obtener las localizaciones más probables. Si analizamos detenidamente la transformada de Hough, podemos observar que realmente estamos proponiendo un algoritmo para realizar dicho cálculo de forma más eficiente.

Como hemos podido ver en los diversos algoritmos que hemos estudiado, la forma en que se realiza la transformación al espacio de parámetros es mediante la acumulación de evidencias. En cada punto de este espacio, que corresponde a una curva C del espacio de la imagen, aparecerá el número de puntos frontera que se sitúan en dicha curva. Por tanto, el número de evidencias que se tendrá en una localización coincide con el resultado de la correlación de la curva en esa localización.

Como indicábamos, el uso de la operación de correlación sobre todas las posibles curvas requeriría una cantidad de tiempo demasiado alta. ¿Cuál es el valor añadido de la transformada de Hough?

Para una determinada localización de la curva, el conjunto de operaciones a realizar para obtener el valor de correlación consiste en la multiplicación punto a punto de los valores de la máscara por los correspondientes de la imagen. Ahora bien, La mayoría de los puntos de la máscara son cero, al igual que ocurre para la mayoría de los puntos de la imagen. Por tanto, sólo un pequeño porcentaje de las operaciones que se llevan a cabo son realmente útiles en el sentido de que afectan al valor o conjunto de votos de esa posición. La eficacia de la transformada de Hough surge del hecho de que considera sólo estas operaciones, es decir, recorre cada uno de los puntos frontera y cada uno de los puntos que forman la curva de manera que en cada paso realiza una de las operaciones válidas, es decir, que provocan un incremento de un voto en el resultado.

9.3.2. Requisitos de espacio y tiempo.

Uno de los grandes inconvenientes que hacen difícil la utilización de la transformada de Hough en la práctica es la gran cantidad de espacio y tiempo que requiere.

Los recursos en espacio vienen directamente determinados por el tamaño del acumulador que se utiliza. Dado que se utiliza una matriz p -dimensional $A[V_1, \dots, V_p]$ de dimensiones $n_{V_1} \times \dots \times n_{V_p}$ (véase sección 9.2.3), los requisitos de espacio serán

$$O(n_{V_1} \times \dots \times n_{V_p}) \quad (9.24)$$

Estos recursos crecerán de forma exponencial con el número de parámetros que inclu-

yamos.

Por otro lado, si analizamos el algoritmo para detección de curvas con expresión analítica de la sección 9.2.3, podemos ver que el algoritmo consta básicamente de dos partes: votación y extracción de máximos.

Con respecto a la votación, para cada punto frontera de la imagen se deben recorrer todos los valores de V_1, \dots, V_{p-1} y para cada uno de ellos se realiza una operación de votación de eficiencia $O(1)$. Por tanto, si tenemos n puntos frontera, la eficiencia será de

$$O(n \times n_{V_1} \times \dots \times n_{V_{p-1}}) \quad (9.25)$$

que también crece de forma exponencial con el número de parámetros que incluyamos. Nótese que usar las ecuaciones paramétricas nos permite eliminar uno de estos parámetros pero a cambio tenemos que incluir el parámetro t que se mueve en el intervalo $[0, 2\pi]$. En este caso, si el número de muestras de este intervalo es n_t la eficiencia será

$$O(n \times n_{V_1} \times \dots \times n_{V_{p-2}} \times n_t) \quad (9.26)$$

Con respecto a la extracción de máximos, necesitaremos estudiar el espacio de parámetros para extraer los puntos de mayor acumulación de votos. Aunque en principio esta operación podría simplemente un recorrido por todas sus posiciones y la correspondiente comprobación, en la práctica necesita de algunas operaciones más. Esto se debe principalmente a los problemas de discretización y errores en el cálculo de fronteras que afectan directamente a la concentración y dispersión de votos. Así, la obtención de máximos se realiza primero con alguna operación de alisado que haga al algoritmo más robusto a estos problemas, y segundo con la extracción propiamente dicha. Una vez seleccionado un punto es conveniente eliminar el máximo junto con los valores vecinos para seleccionar el siguiente.

Por lo tanto, la eficiencia en esta fase vendrá afectada por las dimensiones del acumulador y por los métodos de postprocesado que se consideren. Por ejemplo, si realizamos un alisado con un filtro de 3^p elementos en el dominio espacial, la eficiencia del alisado será

$$O(3^p \times n_{V_1} \times \dots \times n_{V_p}) \quad (9.27)$$

En definitiva, el tiempo requerido para esta fase dependerá también de forma exponencial del número de parámetros.

Resumiendo, los factores que afectan directamente a la eficiencia del algoritmo son:

- El número de parámetros.
- El número de muestras de cada parámetro.
- El número de puntos frontera obtenidos.

A fin de aliviar estos requerimientos y hacer más recomendable la utilización de la transformada de Hough, se han propuesto multitud de modificaciones y algoritmos en la literatura. En la siguiente sección vamos a ver un método para mejorar la eficiencia.

9.3.3. Mejora de eficiencia: Uso del gradiente.

Hasta ahora sólo se ha usado como evidencia la situación de los puntos frontera en el espacio de la imagen. Una información que hemos obviado y que resulta muy interesante al mejorar la eficiencia de los algoritmos es el uso del gradiente.

Cuando se procesa un punto frontera, se realiza una votación en todas aquellas posiciones del espacio de parámetros que pudieran dar lugar a una curva que pasa por ese punto. Si tenemos en cuenta la dirección del gradiente, dado un punto y una dirección, sólo votaremos en las posiciones que dan lugar a una curva que pasa por ese punto **y además tienen ese mismo gradiente**.

Desde otro punto de vista, la localización de un punto da lugar a una condición (ecuación) que nos fija un parámetro a partir de los demás. El incluir el gradiente permite tener en cuenta una nueva condición por lo que podremos fijar dos parámetros a partir de lo demás. En términos de eficiencia el uso de esta información implica, por tanto, disminuir en uno las dimensiones a tener en cuenta durante la votación.

Gradiente aplicado a la detección de líneas rectas.

El algoritmo de votación para la detección de líneas rectas necesitaba recorrer todos los valores de θ y para cada uno de ellos, calcular ρ para acumular un voto. Dado que podemos considerar que la dirección del gradiente es perpendicular a la recta que buscamos, si calculamos el ángulo que forma el vector gradiente y el eje de abscisas, obtenemos directamente el parámetro θ de dicha recta. Este valor se puede usar en la ecuación (9.8) y obtener directamente el valor de ρ .

El algoritmo quedaría

Algoritmo de detección de rectas usando gradiente

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en I .
 2. Inicializar el acumulador $A[\theta, \rho]$ a cero.
 3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) $\theta = \text{Angulo}(\text{grad}_x(x, y), \text{grad}_y(x, y))$
 - b) $\rho = x \cos \theta + y \sin \theta$
 - c) $A[\theta, \rho] = A[\theta, \rho] + 1$
 4. Los máximos del acumulador $A[\theta, \rho]$ corresponden a los puntos frontera que se encuentran alineados en la imagen I .
-

Por lo tanto sólo es necesario un voto por cada punto frontera.

Gradiente aplicado a la detección de círculos.

Para realizar la votación en la detección de círculos, para cada punto frontera recorremos todos los valores del radio y todos los valores del parámetro t que recorre la curva. Al añadir el gradiente, y por tanto una condición más, sólo necesitamos recorrer los distintos valores del radio y determinar para cada uno de ellos el centro del círculo correspondiente⁴.

El algoritmo por tanto quedaría como sigue:

Algoritmo de detección de círculos usando gradiente

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
 2. Para todo (a, b, r) tal que $a \in \{0, \dots, N-1\}$, $b \in \{0, \dots, N-1\}$, $r \in \{1, \dots, n_r-1\}$ hacer
 - a) $A[a, b, r] = 0$
-

⁴Nótese que aquí usamos no sólo la dirección sino también el sentido del gradiente. Si el sentido no nos determina la posición del objeto tendríamos que votar en dos posiciones distintas

3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Para todo (r) tal que $r \in \{1, \dots, n_r - 1\}$ hacer
 - 1) $t = \text{Angulo}(\text{grad}_x(x, y), \text{grad}_y(x, y))$
 - 2) $a = x - r \cos t$
 - 3) $b = y - r \sin t$
 - 4) $A[a, b, r] = A[a, b, r] + 1$
 4. Los máximos del acumulador $A[a, b, r]$ corresponden a las curvas en la imagen I .
-

Gradiente aplicado a la detección de una curva cualquiera.

Como se ha indicado anteriormente, podemos construir un algoritmo para localizar una curva de p parámetros usando las ecuaciones paramétricas. Este algoritmo nos permitía definir directamente la forma en que podíamos localizar una curva dibujada pues básicamente el bucle interior sólo tenía que dibujar la curva (véase la sección 9.2.3, página 199).

El uso del gradiente afecta a este algoritmo eliminando la necesidad de dibujar la curva completa. Cuando no se tenía información de gradiente, hacía falta recorrer cualquier punto de la curva ya que la evidencia podría corresponder a cualquiera de esos puntos. Ahora el gradiente determina que la evidencia sólo puede corresponder a aquellos puntos de la curva cuya tangente sea perpendicular a ese gradiente. Por tanto, el bucle interno que dibujaba la curva, ahora sólo tendrá que representar esos puntos compatibles. Por ejemplo, en el caso de un círculo, para un radio determinado, un punto y su gradiente determinan una única circunferencia (como se ha visto en la sección anterior).

Para una curva cualquiera (considérese por ejemplo la de la figura 9.12) la inclusión del gradiente se puede realizar mediante la inclusión de una R -tabla⁵. La idea consiste en la partición del conjunto de puntos de la curva clasificándolos según su gradiente o la tangente al contorno. Cada uno de estos conjuntos será una entrada en la R -tabla, y para cada una de estas entradas se almacenan un conjunto de valores (r, α) que indican la correspondiente localización del punto de referencia.

⁵El algoritmo propuesto por Ballard inicialmente como la *Transformada de Hough Generalizada* describía el algoritmo en términos de esta R -tabla y contenía los pasos que se describen en esta sección.

La forma de la R – tabla se muestra en la tabla 9.1. Si localizamos un punto frontera con una tangente de valor ϕ_i , esta tabla nos lleva al conjunto de puntos que hay que votar, es decir, al primer punto de radio r_i^1 y dirección α_i^1 , al segundo punto de radio r_i^2 y dirección α_i^2 , hasta el último punto n_i .

ϕ_1	\rightarrow	(r_1^1, α_1^1)	(r_1^2, α_1^2)	\cdots	$(r_1^{n_1}, \alpha_1^{n_1})$
ϕ_2	\rightarrow	(r_2^1, α_2^1)	(r_2^2, α_2^2)	\cdots	$(r_2^{n_2}, \alpha_2^{n_2})$
ϕ_3	\rightarrow	(r_3^1, α_3^1)	(r_3^2, α_3^2)	\cdots	$(r_3^{n_3}, \alpha_3^{n_3})$
\cdots	\rightarrow	\cdots	\cdots	\cdots	\cdots
ϕ_k	\rightarrow	(r_k^1, α_k^1)	(r_k^2, α_k^2)	\cdots	$(r_k^{n_k}, \alpha_k^{n_k})$

Cuadro 9.1: R-Tabla

En la figura 9.14 se muestra gráficamente el sentido de estos valores. Se puede observar como dos puntos de la curva tienen una misma tangente ϕ y sus dos parejas de valores (r_i, α_i) que indican la posición del centro.

En una aplicación que busque esta curva, si localizamos un punto que corresponda a una tangente ϕ , tendremos que votar en dos posiciones de desplazamiento, que podrán ser localizadas con esas dos parejas de valores. El algoritmo para obtener el desplazamiento se muestra a continuación:

Algoritmo de desplazamiento de una curva dibujada

1. Obtener \mathcal{P}_I , el conjunto de puntos que marcan discontinuidades en la imagen original I .
2. Se inicializa el acumulador A a cero.
3. Para todo $(x, y) \in \mathcal{P}_I$ hacer
 - a) Determinar el valor ϕ de tangente en el punto (x, y)
 - b) Para todo valor (r, α) de la entrada ϕ de la R-Tabla hacer
 - 1) $a = x + r \cos \alpha$
 - 2) $b = y + r \sin \alpha$
 - 3) $A[a, b] = A[a, b] + 1$
4. Los máximos del acumulador $A[a, b]$ corresponden a las curvas en la imagen I .

Nótese en este algoritmo como la nueva aportación es que el bucle interior no dibuja toda la curva, sino que utiliza la partición determinada por la R-tabla para dibujar únicamente aquellos puntos que tienen una pendiente determinada.

Errores en el gradiente.

El uso de la información de gradiente para determinar un subconjunto de posiciones en donde votar puede llevarnos a resultados incorrectos si no tenemos en cuenta el posible error en la estimación de la dirección. Un pequeño error en el valor del ángulo puede implicar un desplazamiento en la posición del voto de varios píxeles con respecto a la posición buscada.

Consideremos el caso de la estimación de θ para aplicarla a la ecuación normal de la recta:

$$\rho = x \cos \theta + y \sin \theta \quad (9.28)$$

Si el ángulo es cercano a $\pi/2$ y el valor de x es muy alto o el ángulo es cercano a 0 y el valor de y es alto, el error en la estimación de ρ puede ser del orden de varias unidades.

Si consideramos por otro lado el caso del círculo, si el radio es suficientemente grande, un error pequeño en la estimación del ángulo del gradiente puede desplazar varias posiciones el centro del círculo.

En la figura 9.15 se muestran gráficamente estas dos situaciones.

Por lo tanto, si el error en el cálculo del gradiente puede ser bastante alto, tendremos que evitar el desplazamiento que sufren los votos. En la práctica, la solución a este problema puede ser el votar no sólo en el ángulo detectado sino en un entorno de éste. Por ejemplo, para evitar el problema en la detección del círculo, no sólo se vota en el centro (a, b) sino en un pequeño arco centrado en ese punto.

9.4. Ejercicios.

9.4.1. Transformada de Hough para localizar líneas rectas.

Implementar el algoritmo de localización de líneas rectas usando la parametrización $\rho - \theta$. Las entradas del programa serán:

- Una imagen de niveles de gris.
- Un método de extracción de máximos. Se proponen dos posibilidades:
 - Especificar el número de rectas a extraer.
 - Indicar el mínimo número de votos para obtener una recta.

Como salidas deberá tener:

- Imagen de fronteras utilizada.
- Imagen con las rectas dibujadas.
- Imagen de niveles de gris correspondiente al espacio de parámetros.

9.4.2. Uso del módulo del gradiente.

A lo largo del tema se ha estudiado el uso de un conjunto de puntos frontera para obtener el resultado de la transformada de Hough. A esta información se le ha añadido la dirección del gradiente para mejorar la eficiencia del algoritmo. Considerar la posibilidad de usar el módulo del gradiente en la transformada. ¿Qué ventajas podrían obtenerse?. Comentar los cambios que habría que realizar para incluir esta información en el algoritmo.

9.4.3. Detección de segmentos.

La localización de rectas en una imagen sufre del problema de que pueden obtenerse segmentos de recta muy distantes en localización aunque de la misma recta. Diseñar un algoritmo para detección de segmentos en una imagen.

9.4.4. Unión de fronteras.

Diseñar e implementar un algoritmo para la unión de fronteras por medio de la transformada de Hough.

9.4.5. Uso del gradiente en la detección de una curva cualquiera con rotaciones.

En la sección 9.3.3 se ha comentado el uso del gradiente para la detección de una curva cualquiera y se ha propuesto un algoritmo para obtener el desplazamiento. Comente y diseñe el algoritmo de detección usando además escala y rotación (véase sección 9.2.3).

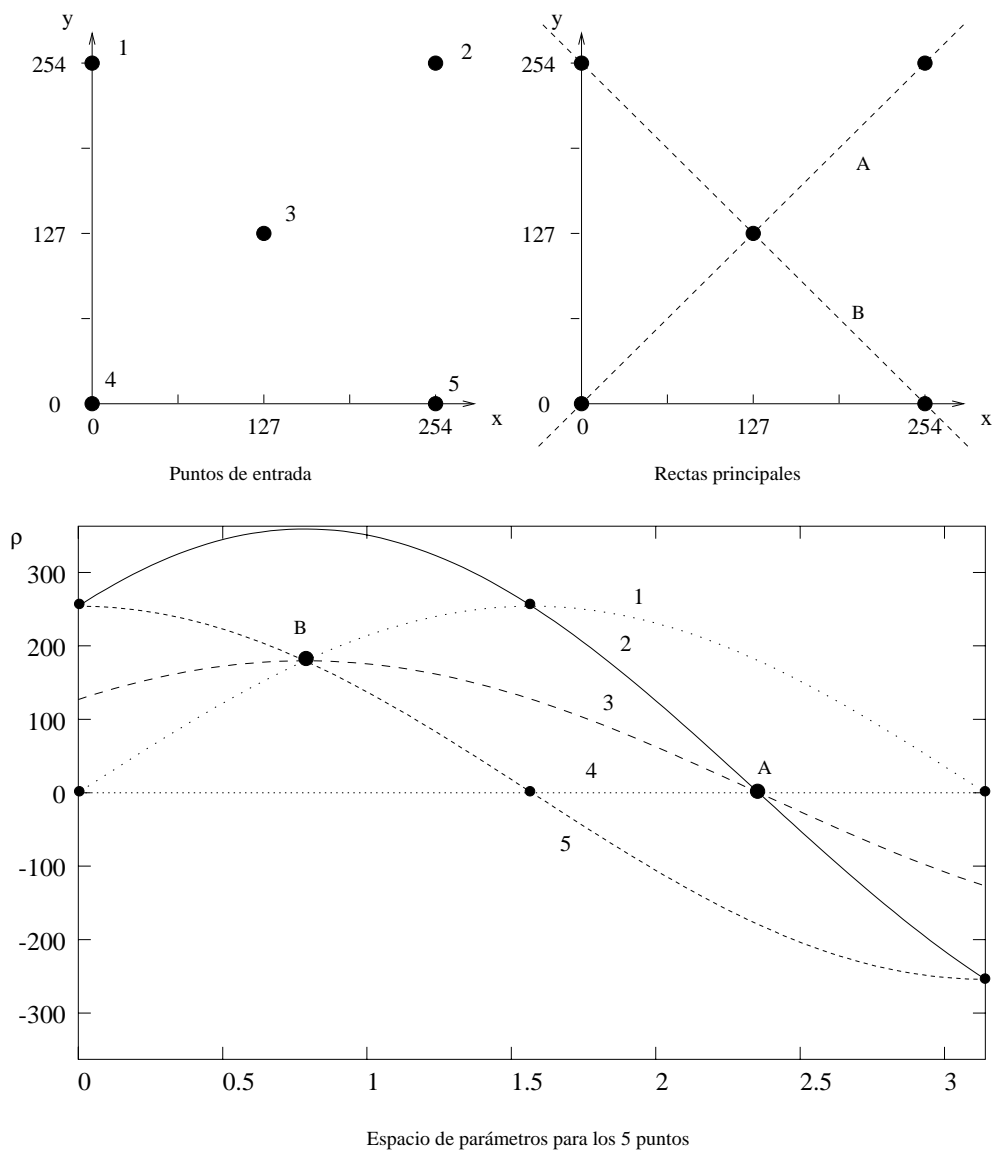


Figura 9.6: Transformada de Hough usando el espacio $\theta - \rho$.

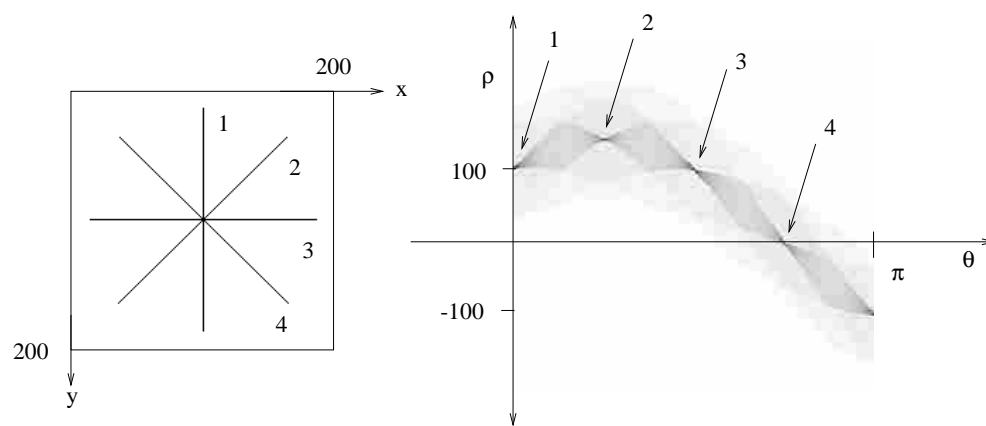


Figura 9.7: Ejemplo sencillo de 4 líneas rectas.

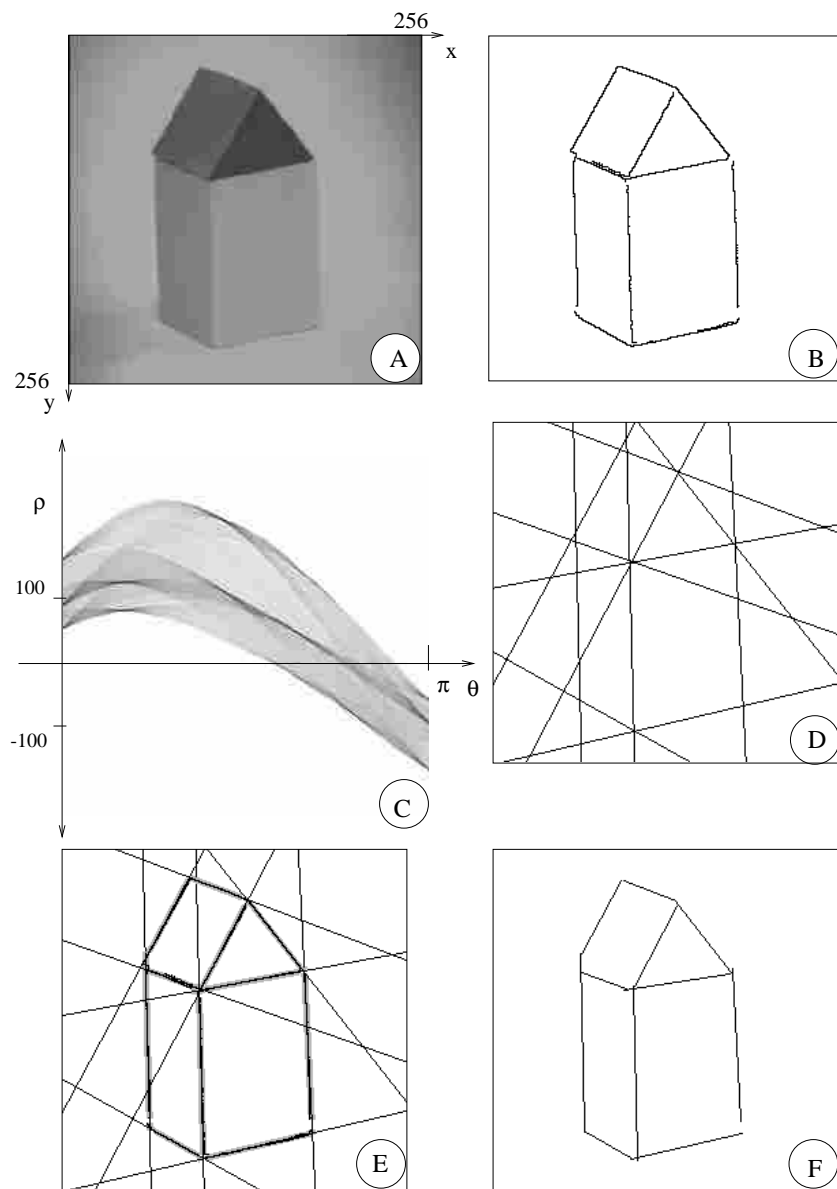


Figura 9.8: Detección de líneas de una imagen real.

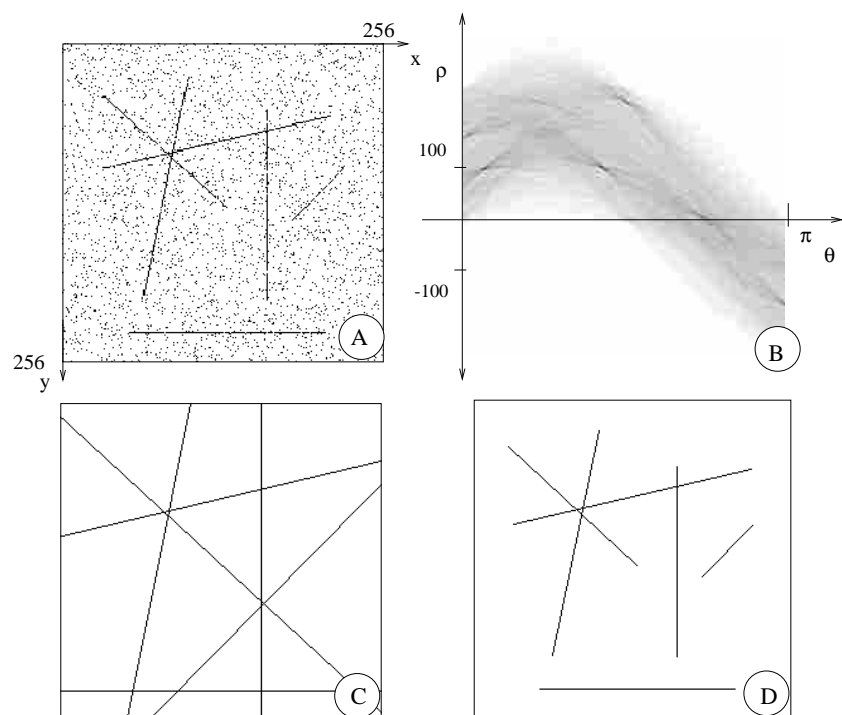


Figura 9.9: Detección de líneas en una imagen ruidosa.

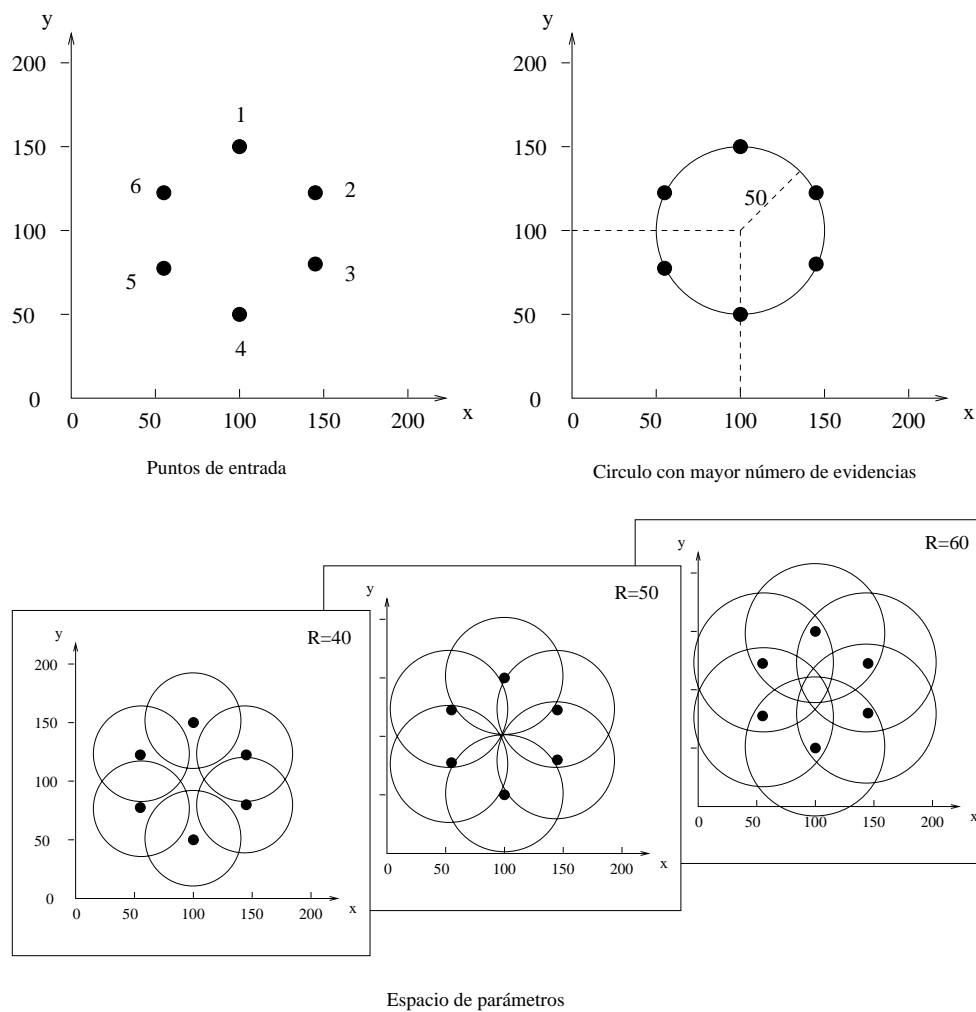


Figura 9.10: Transformada de Hough para la localización de círculos.

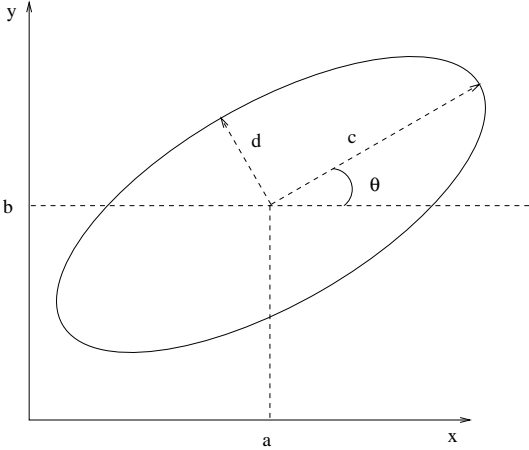


Figura 9.11: Parametrización de una elipse.

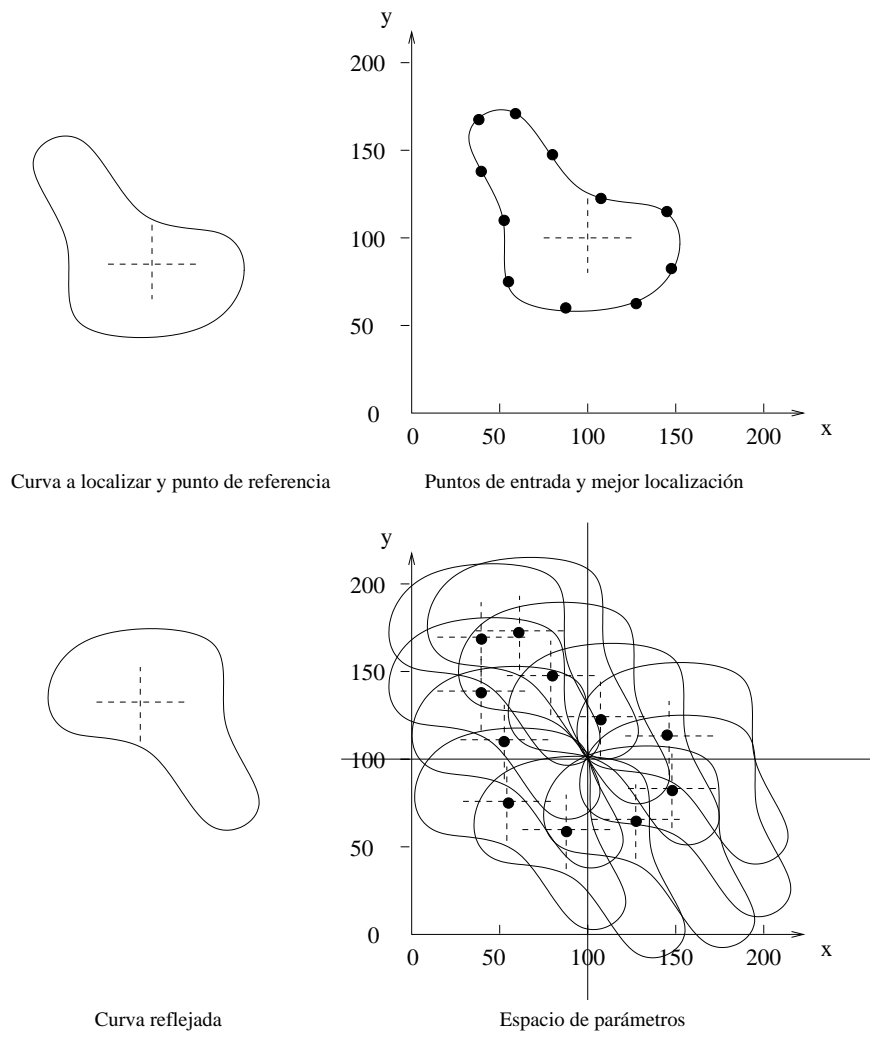
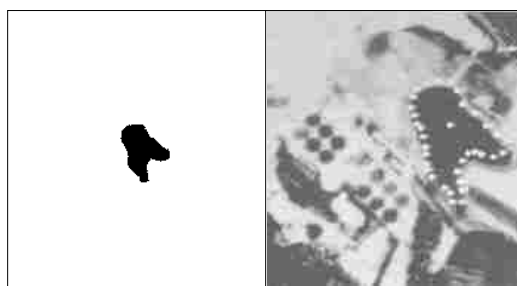


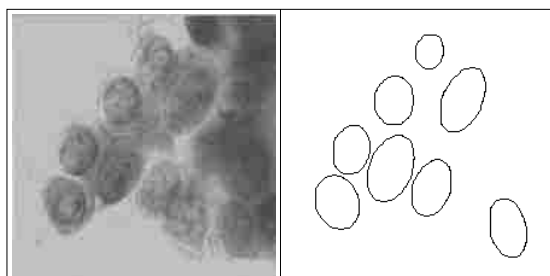
Figura 9.12: Transformada de Hough para la localización de una curva cualquiera.



(a) Detección de Círculos



(b) Generalización a una curva dibujada



(c) Detección de elipses

Figura 9.13: Ejemplos de generalización.

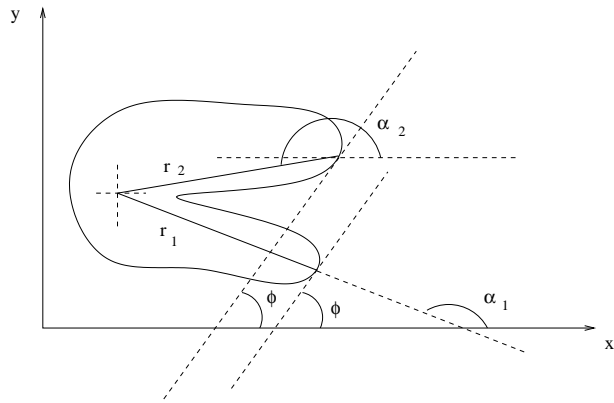


Figura 9.14: Geometría de la R-Tabla en la Transformada de Hough Generalizada.

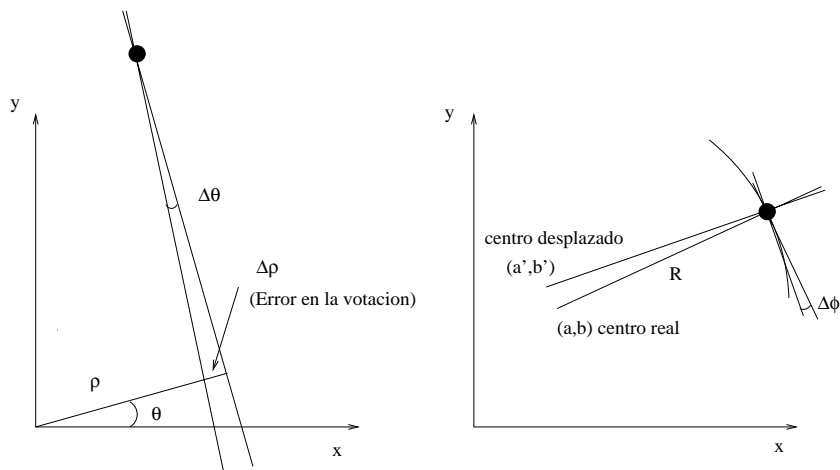


Figura 9.15: Errores en el gradiente.

Capítulo 10

Representación y Descripción de Fronteras y Regiones

10.1. Introducción

Una vez que una imagen ha sido segmentada en regiones por los métodos discutidos en los temas anteriores, es necesario representar y describir dichos píxeles de forma que puedan ser procesados posteriormente. Podría decirse que existen dos opciones:

1. Representar una región en función de su frontera,
2. realizar la representación en función de sus características internas.

Una vez elegida la representación hemos de buscar una forma de describirla. Por ejemplo, nos podemos decidir por una representación basada en la frontera y describir la frontera mediante su longitud, el número de concavidades que tiene, etc.

Podría decirse que la representación basada en la frontera se utiliza más cuando el énfasis es en la forma. La representación basada en la región se usa más cuando se buscan propiedades como color y textura. En cualquier caso, los rasgos que hemos de seleccionar como descriptores deberían ser insensibles a cambios como variaciones de tamaño, traslaciones, rotaciones, etc. Las referencias básicas de este tema son [13, 21, 48]

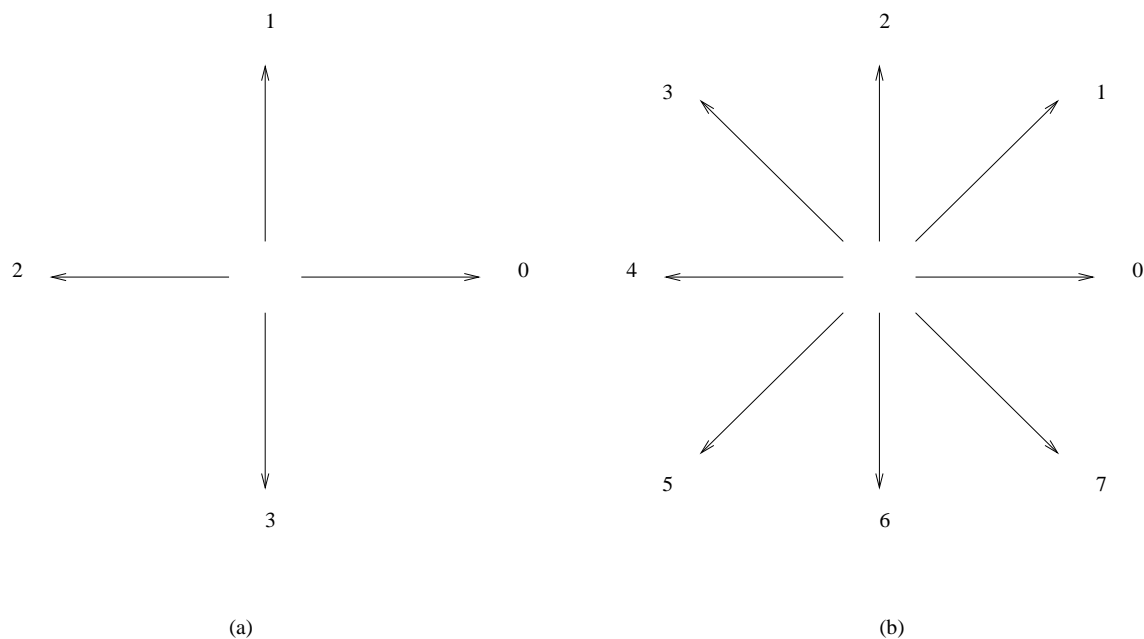


Figura 10.1: Direcciones utilizadas para codificar una frontera

10.2. Representaciones de la Frontera

10.2.1. Cadenas de códigos

Las cadenas de códigos se utilizan para representar fronteras mediante una sucesión conectada de segmentos de líneas rectas de una longitud y dirección dada. Normalmente esta representación se basa en la 4- u 8- conectividad. La dirección de cada segmento se codifica mediante un esquema de numeración como los que se muestran en la figura 10.1.

La figura 10.2 muestra un ejemplo de representación de una frontera utilizando cadenas de códigos. Obviamente, la cadena de códigos depende del punto de partida y es necesaria su normalización para una futura comparación. Esta normalización se puede alcanzar (ver [39] para detalles).

10.2.2. Aproximaciones Poligonales

Una frontera digital puede ser aproximada mediante un polígono. Si la curva es cerrada, la aproximación es exacta cuando el número de segmentos en el polígono coincide con el

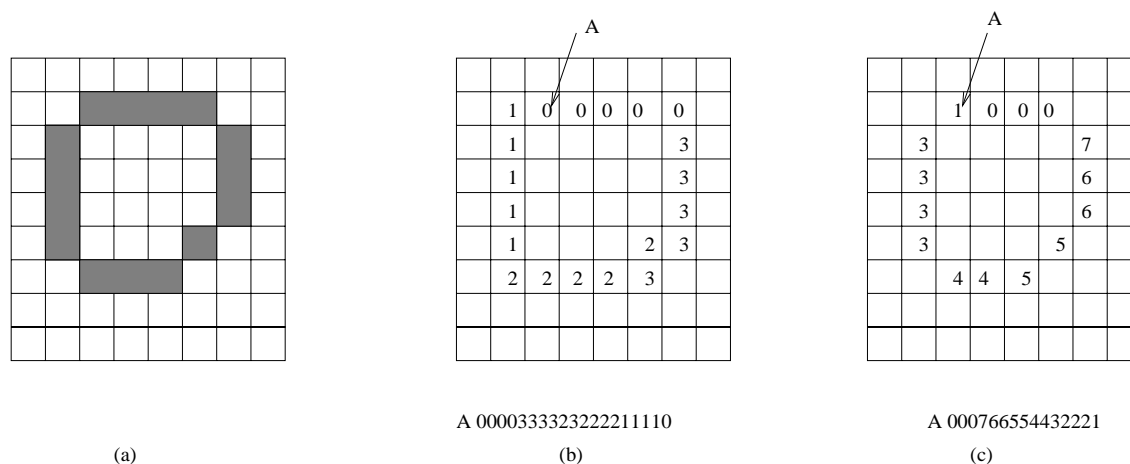


Figura 10.2: Codificaciones de una frontera. (a) Frontera, (b) 4-conectividad, (c) 8-conectividad.

número de puntos en la frontera, de forma que cada par de puntos adyacentes define un segmento del polígono. En la práctica el objetivo de una aproximación poligonal es capturar la esencia de la forma de la frontera con el menor número posible de segmentos poligonales. El problema no es trivial y puede necesitar una enorme potencia de cálculo. Sin embargo, varias técnicas de aproximación poligonal son al mismo tiempo simples y medianamente buenas.

Una aproximación frecuentemente utilizada subdivide un segmento sucesivamente en dos partes hasta que se cumpla un determinado criterio. Por ejemplo, una exigencia podría ser que la distancia perpendicular máxima desde un segmento de frontera a la línea que une sus dos puntos finales no exceda un umbral dado. Si excede el umbral, el punto más alejado se convierte en vértice y divide el segmento en dos subsegmentos. Esta aproximación tiene la ventaja de que busca puntos de inflexión prominentes. Para una frontera cerrada, los puntos de comienzo mejores son los dos más alejados. El proceso se ilustra con un ejemplo en la figura 10.3.

10.2.3. Signatura

Una signatura es una representación funcional $1 - D$ de una frontera y puede ser generada de varias formas. La mas sencilla es dibujar la distancia desde el centroide a la

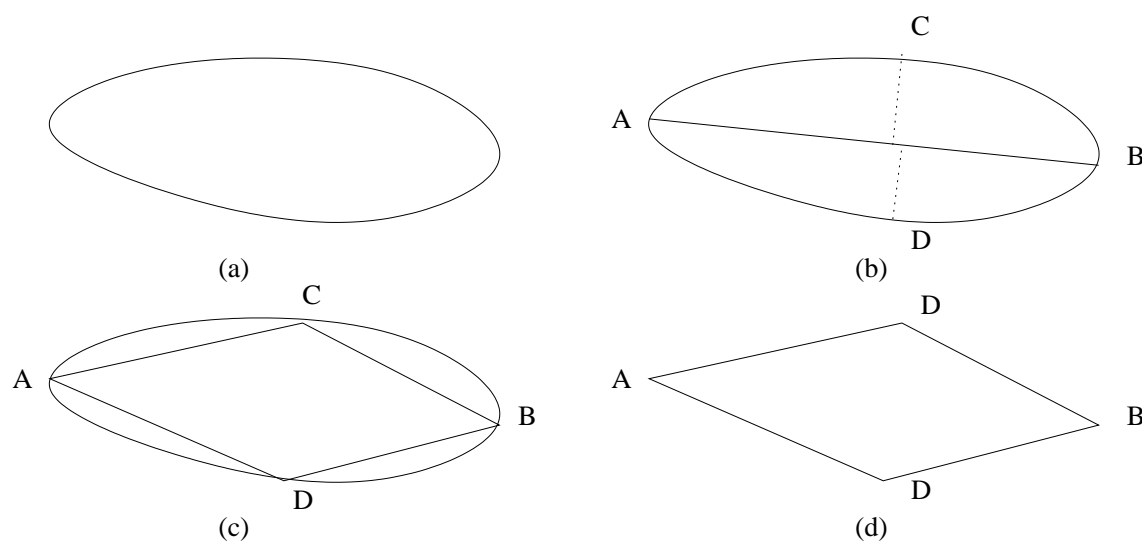


Figura 10.3: Representando una frontera por aproximaciones poligonales.

frontera en función del ángulo, como se ve en la figura 10.4. Independientemente de como sea generada la signatura el objetivo es reducir la frontera a una representación unidimensional, que en principio será más fácil de describir que la frontera original bidimensional.

El proceso descrito para generar la signatura es invariante por traslaciones pero no es invariante con relación a la escala y rotación. Para conseguir la normalización por rotación la idea puede ser comenzar siempre por el mismo punto para generar la signatura. El punto puede ser el de distancia máxima al centroide. Para la normalización con relación a la escala la idea puede ser llevar todos los valores al intervalo $[0, 1]$. Sin embargo, este escalado utilizando el mínimo y el máximo puede generar problemas debido, por ejemplo, al ruido. Una alternativa es realizar el escalado localmente teniendo en cuenta la varianza de la parte de la frontera donde nos encontremos.

La representación distanciaángulo es sólo una de las posibilidades. Podemos utilizar, entre otros la función de densidad de la pendiente como signatura. Puesto que un histograma es una medida de concentración de valores, la densidad de la frontera responde fuertemente a secciones de la frontera con ángulos tangentes constantes (segmentos rectos o casi rectos) y tiene mínimos, normalmente, en secciones con ángulos que varían rápidamente.

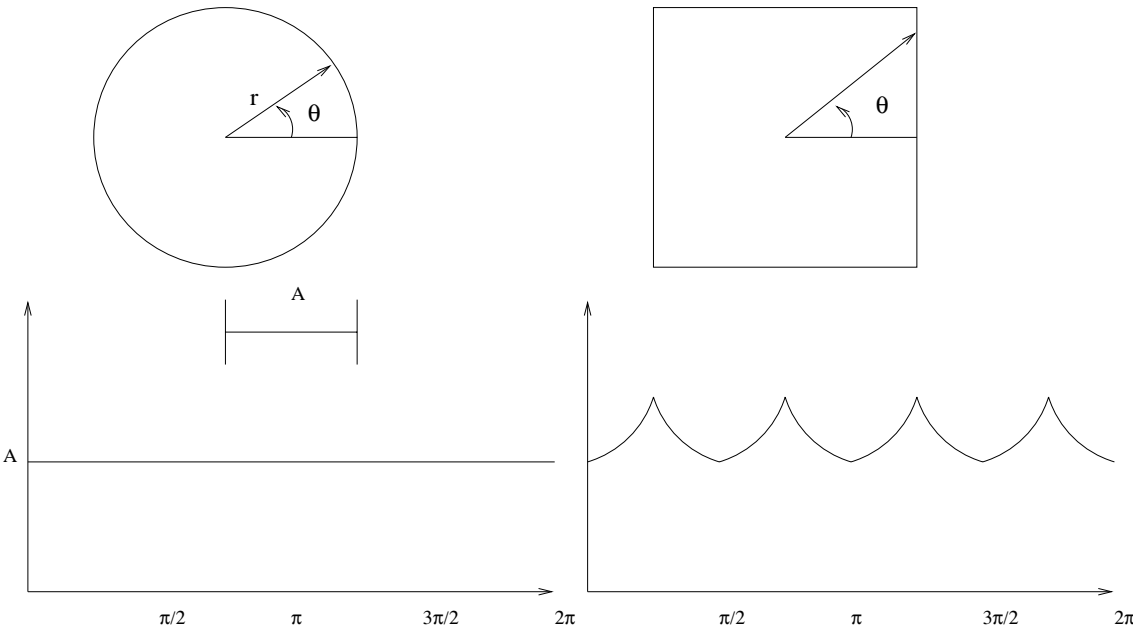


Figura 10.4: Signatura de dos fronteras distintas.

10.3. Representaciones de las regiones

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

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

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

10.3.1. Códigos de Longitud Variable

Cualquier región o imagen binaria puede ser vista como una sucesión en la que se alternan hileras de 0s y 1s. Los códigos de longitud variable representan estas hileras. Para una imagen que es escaneada por filas hasta llegar a la región de interés, el código consiste en la dirección de comienzo de cada hilera seguida del número de 1s o 0s que la integran. Un ejemplo se encuentra en la figura 10.5

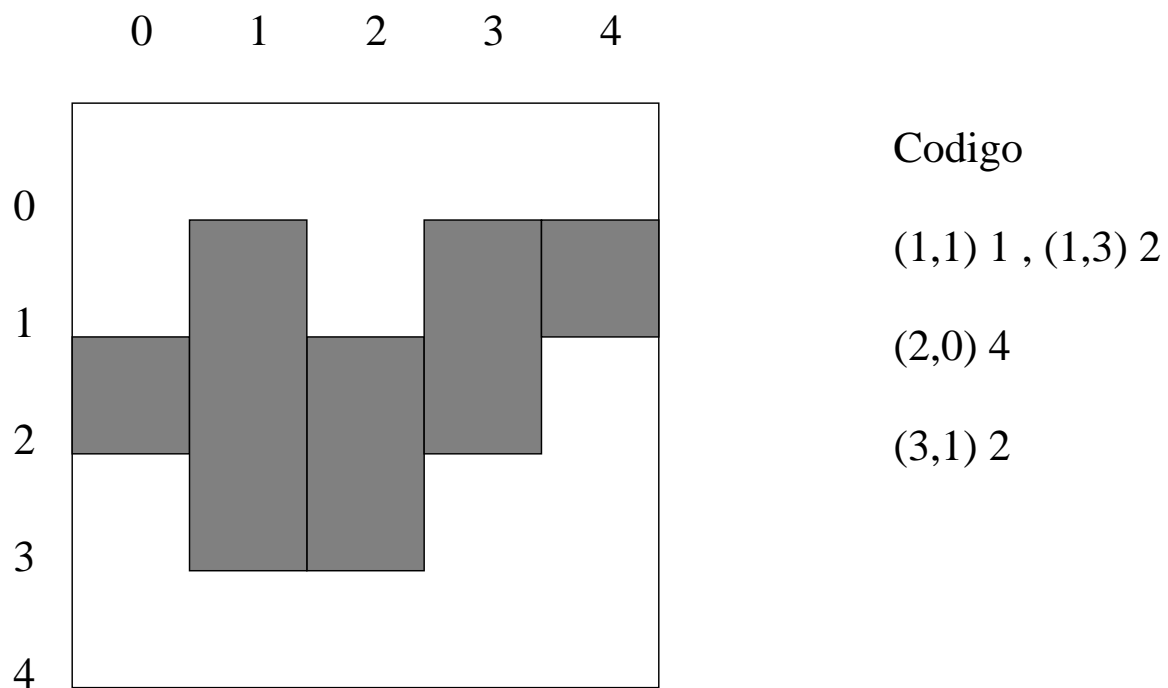
10.3.2. Quad-trees

En este método, la región dada se incluye en un área rectangular conveniente. Este área es dividido en cuatro cuadrantes y se examina cada uno de ellos para ver si todos sus píxeles pertenecen a la misma región o no. Si tiene pertenecientes y no pertenecientes a la región, se sigue subdividiendo. Una estructura de árbol se genera hasta que cada subcuadrante sólo contiene píxeles de un tipo. Supongamos que utilizamos el 1 para la región y el 0 para la no región, entonces el árbol puede codificarse de forma única como una hilera de 0s, 1s y gs para denotar no región, región y subdivisión necesaria, ver figura 10.6.

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

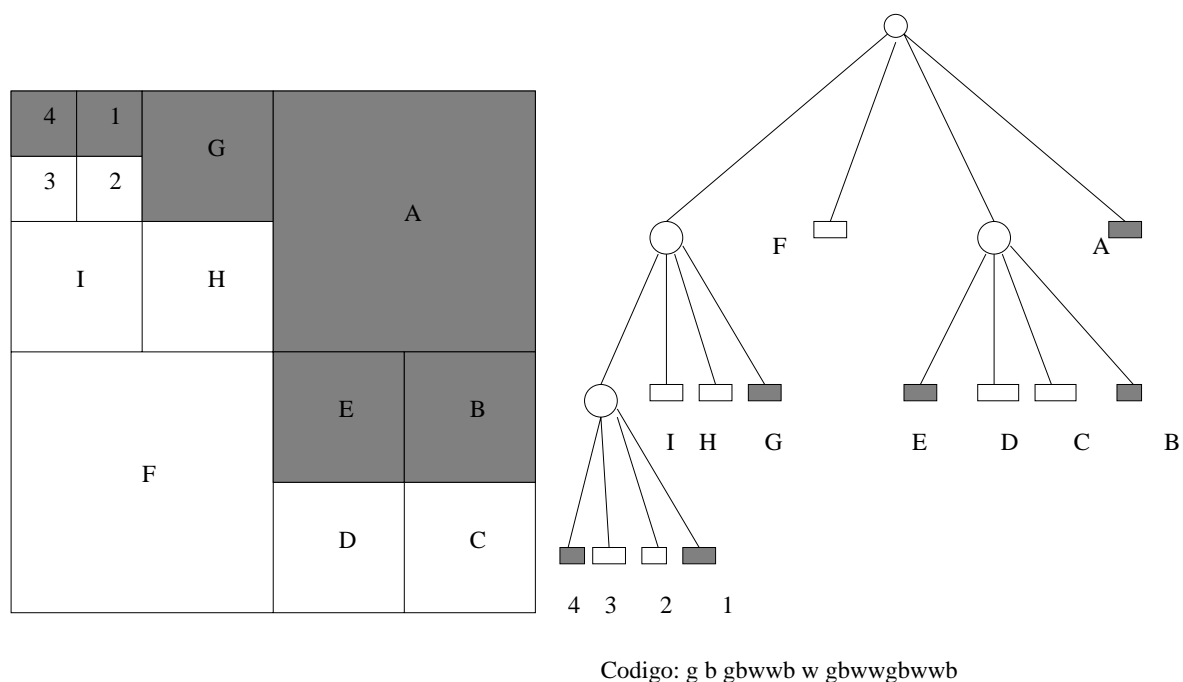
10.3.3. Proyecciones

Una forma bidimensional o región \mathcal{R} puede ser representada también por sus proyecciones. Una proyección $g(s, \theta)$ es simplemente la suma de las longitudes de las hileras a lo



(a) Imagen binaria

Figura 10.5: Codificaciones de una región. (a) Imagen binaria y su codificación.



(a) Cuadrantes

(b) Creando el quad-tree

Figura 10.6: (a) Región, (b) Quad-tree y su representación

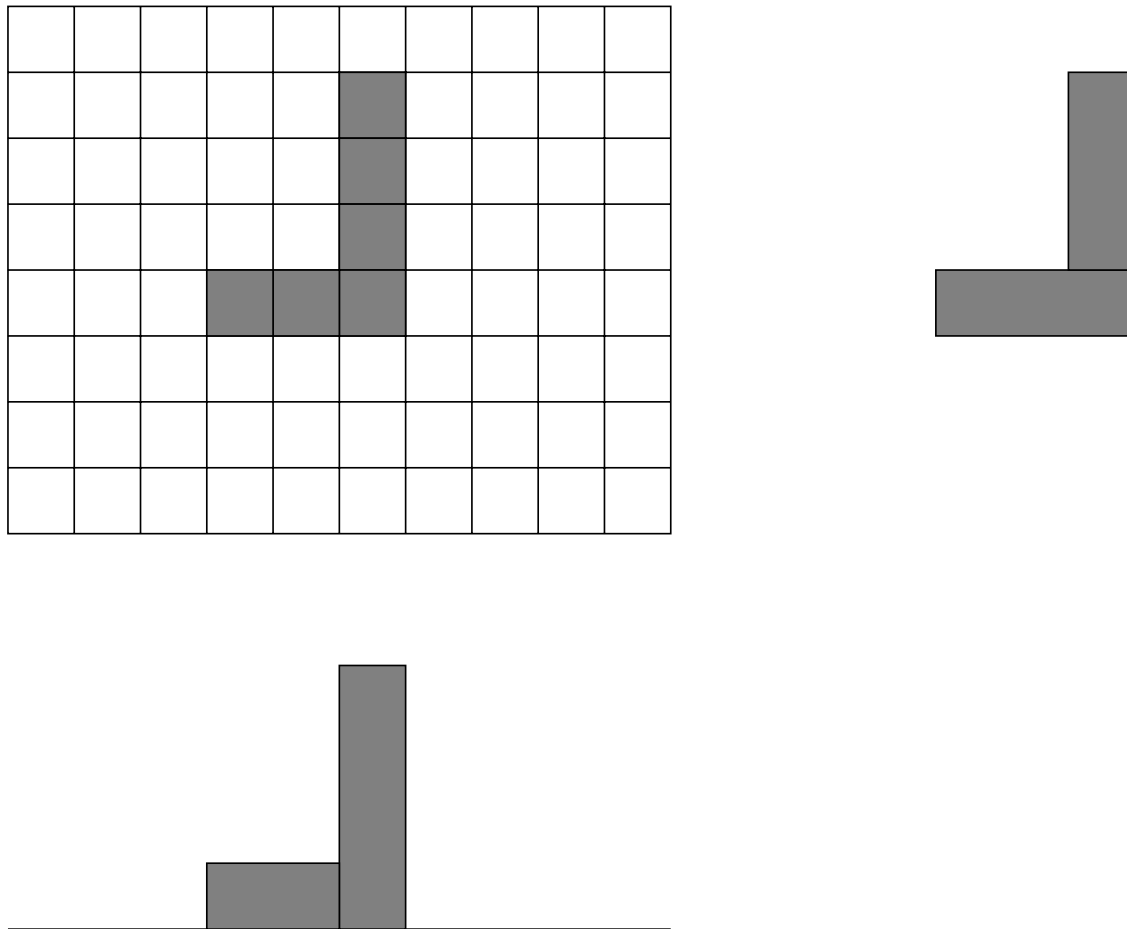


Figura 10.7: Una imagen y dos de sus proyecciones.

largo de una línea recta orientada a un ángulo θ y a una distancia s . En este sentido una proyección es un histograma que proporciona el número de píxeles que se proyectan en una determinada posición a una distancia s en la línea de orientación θ (ver figura 10.7).

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

10.4. Descriptores de Fronteras

Una vez que tenemos una representación más o menos apropiada de una frontera, hemos de proceder a su descripción. Comencemos con los descriptores más simples.

10.4.1. Algunos descriptores simples

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

El diámetro de una frontera B se define como

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

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

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

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

10.4.2. Descriptores de Fourier

Consideremos la imagen digital que se muestra en la figura 10.8. Si comenzamos en un punto arbitrario (x_0, y_0) , tendremos la sucesión de coordenadas $(x_0, y_0), \dots, (x_{N-1}, y_{N-1})$.

Cada par de coordenadas puede tratarse como un número complejo de forma que tendremos

$$s(k) = x(k) + iy(k)$$

para $k = 0, 1, \dots, N - 1$.

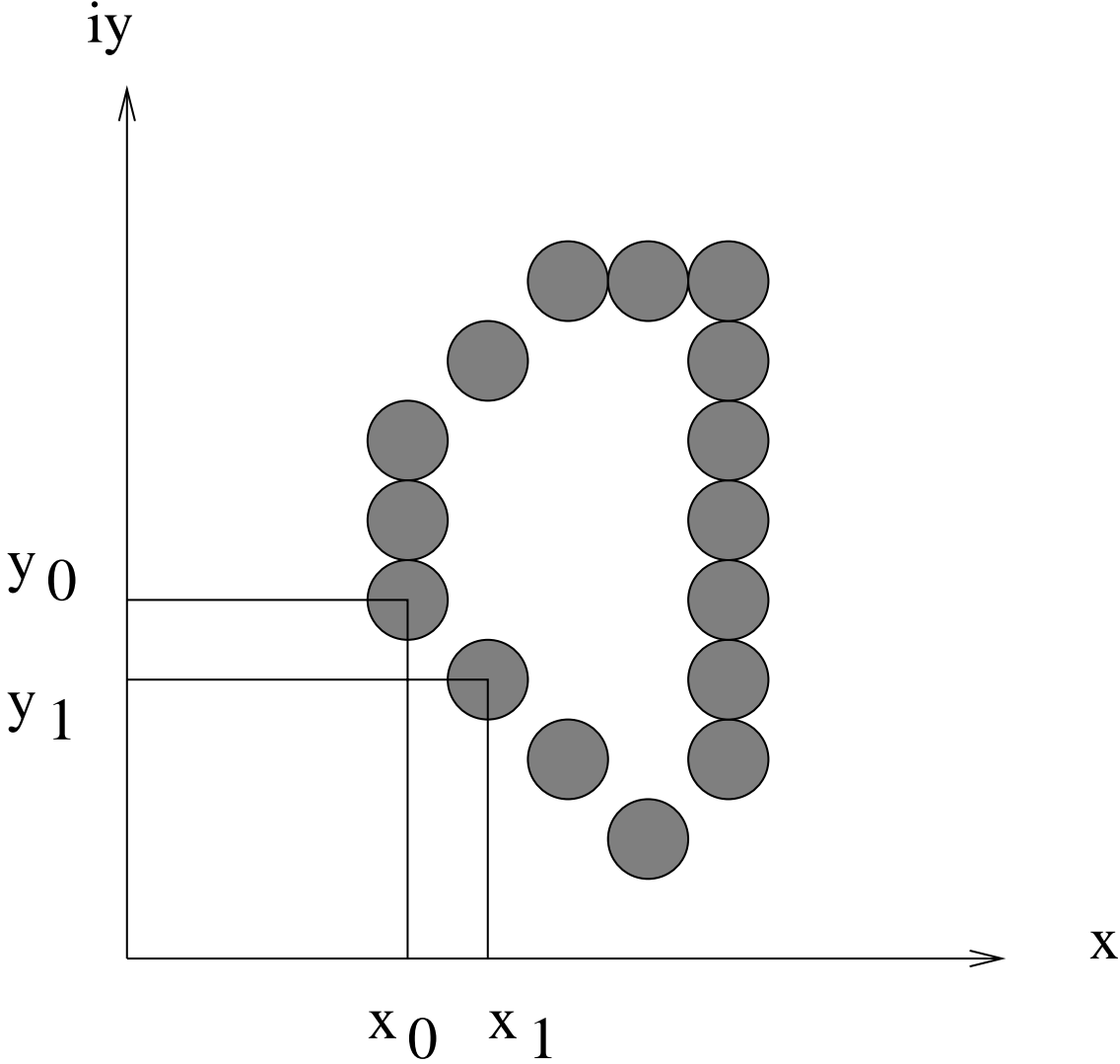


Figura 10.8: Frontera digital y su representación compleja

Obviamente, podemos calcularle la transformada de Fourier a la sucesión $s(k)$ y tendremos

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \exp[-2\pi i u k / N] \quad u = 0, \dots, N-1$$

Los coeficientes complejos $a(u)$ reciben el nombre de descriptores de Fourier de la frontera, y obviamente mediante la transformada inversa podemos recuperar la frontera original.

Supongamos que en lugar de los $a(u)$ sólo utilizamos los M primeros coeficientes y calculamos la transformada inversa. Es decir, escribimos

$$\hat{s}(k) = \sum_0^{M-1} a(u) \exp[2\pi i u k / N] \quad k = 0, \dots, N-1$$

es obvio que estaremos aproximando la frontera. Si el número de puntos en la frontera es grande, M se selecciona normalmente como una potencia de 2. En cualquier caso, lo que hacemos, al suprimir las frecuencias altas en la transformada inversa es quitar los detalles.

Como ya hemos comentado con anterioridad los descriptores deberían ser insensibles, o lo más insensible posible, a traslaciones, rotaciones y cambios de escala, al igual que al cambio de punto de comienzo.

Consideremos una rotación a nuestra frontera y veamos que le pasa a los descriptores de Fourier. La rotación de un punto por un ángulo θ alrededor del origen, se lleva a cabo en el plano complejo mediante la multiplicación por $e^{i\theta}$. La rotación produce entonces la sucesión de números complejos $s(k)e^{i\theta}$, de modo que los descriptores de Fourier son ahora

$$a_r(u) = \frac{1}{N} \sum_0^{N-1} s(k)e^{i\theta} \exp[-2\pi i u k / N] = a(u)e^{i\theta}$$

La figura 10.9 resume como se modifican los descriptores de Fourier de una frontera cuando se lleva a cabo una rotación, escalado, traslación y cambio de puntos de comienzo. El símbolo Δ_{xy} se define como $\Delta_{xy} = \Delta_x + i\Delta_y$, de modo que la notación $s_t(k) = s(k) + \Delta_{xy}$ significa redefinir la sucesión como

$$s_t(k) = x(k) + \Delta_x + i(y(k) + \Delta_y)$$

Finalmente, la expresión $s_p(k) = s(k - k_0)$ significa redefinir la sucesión como

$$s_p(k) = x(k - k_0) + iy(k - k_0)$$

Propiedades básicas de los decriptores de Fourier

Transformación	Frontera	Descirptor de Fourier
Identidad	$s(k)$	$a(u)$
Rotación	$s_r(k) = s(k)e^{i\theta}$	$a_r(u) = a(u)e^{i\theta}$
Traslación	$s_t(k) = s(k) + \Delta_{xy}$	$a_r(u) = a(u) + \Delta_{xy}\delta(k)$
Escalado	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Punto de Inicio	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{i2\pi k_0 u/N}$

Figura 10.9:

10.4.3. Momentos

La forma de una frontera (y su signatura) puede representarse mediante la utilización de momentos. veamos como puede hacerse, consideremos la figura 10.10a que muestra un segmento de frontera, y la figura 10.10b que muestra los segmentos representados como una función unidimensional $g(r)$ de una determinada variable r . Si consideramos la amplitud de g como una variablr aleatoria v y formamos su histograma $p(v_i), i = 1, 2, \dots, K$ donde K es el número de valores que toma la variable, entonces podemos definir los diferentes momentos

$$\nu_n(v) = \sum_{i=1}^K (v_i - m)^n p(v_i)$$

donde

$$m = \sum_{i=1}^K v_i p_i$$

Esta cantidad m puede ser reconocida como una media y ν_2 como la varianza, aunque no estemos trabajando con histogramas normalizados. Podríamos hacer la normalización del histograma y entonces tener ya distribuciones de probabilidad y entonces hablar del momentos.

10.5. Descriptores de Regiones

Una vez analizados los descriptores de fronteras que pueden utilizarse para la clasificación de la misma, procedemos a hacer lo mismo con las regiones.

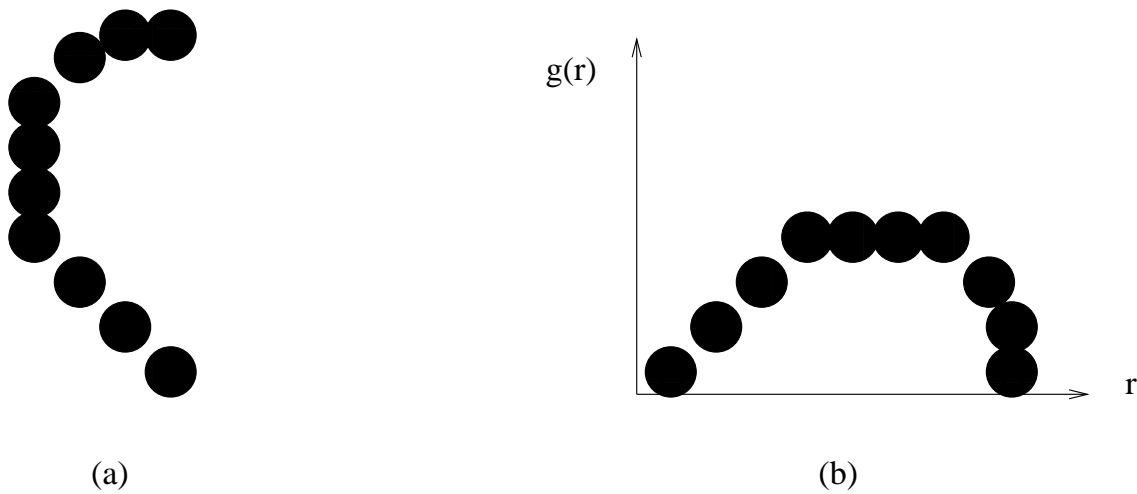


Figura 10.10: (a) Segmento de Frontera, (b) Representación 1-D

10.5.1. Algunos descriptores simples

El *área* de una región se define como el número de píxeles contenidos dentro de su frontera. El *perímetro* de una región es la longitud de su frontera. Aunque tanto el área como el perímetro se utilizan a veces como descriptores, su uso es básicamente en aquellos caso en los que el objeto de interés es invariante.

Un descriptor que se usa más y que surge de los dos anteriores es la *compacticidad* de una región que se define como $\text{perímetro}^2/\text{área}$. Esta es una cantidad sin dimensión y por tanto insensible a cambios de escala, además es mínima para una región con forma de disco. Salvo errores de redondeo debidos a la rotación, la compacticidad es insensible a la orientación.

10.5.2. Momentos

Para una función continua $f(x, y)$, el momento de orden $(p + q)$ se define como

$$m_{pq} = \int \int x^p y^q f(x, y) dx dy$$

Los momentos centrales se definen como

$$\nu_{pq} = \int \int (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy$$

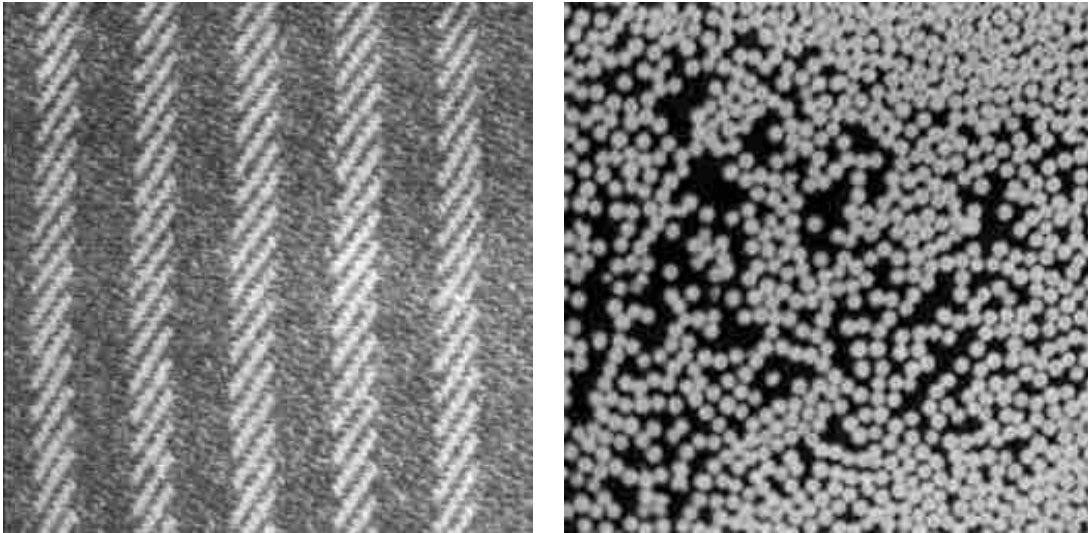


Figura 10.11: Algunos ejemplos de texturas

con

$$\bar{x} = m_{10}/m_{00} \quad \bar{y} = m_{01}/m_{00}$$

Obviamente, para imágenes digitales las integrales se convierten en sumas.

Los momentos centrales normalizados, notados η_{pq} se definen mediante

$$\eta_{pq} = \nu_{pq}/\nu_{00}^\gamma$$

con $\gamma = (p + q)/2 + 1$.

Es importante notar que a partir de los momentos de orden dos y tres se pueden obtener siete momentos invariantes a traslaciones, rotaciones y escalado.

10.5.3. Texturas

Una aproximación importante a la descripción de una imagen es la cuantificación de su contenido en *textura*. Aunque no existe una definición formal de la textura de un objeto, este descriptor proporciona medidas de propiedades como suavidad, regularidad, densidad, ver figura 10.11.

Las tres aproximaciones principales a la descripción por texturas son la estadística, la estructural y la espectral, aquí sólo veremos la primera.

Una de las aproximaciones más simples para describir las texturas es utilizar los momentos del histograma de niveles de gris de la imagen o región.

Así, por ejemplo, podemos calcular la media y la varianza. La varianza, notada $\sigma^2(z)$ donde z se usa para notar nivel de gris, es particularmente importante en la descripción de la textura. Es una medida del contraste de nivel de gris y puede utilizarse para determinar la suavidad de la región. Por ejemplo, la medida

$$R = 1 - \frac{1}{1 + \sigma^2(z)}$$

es cero para áreas de nivel de gris constante y uno para varianzas grandes.

Si utilizamos sólo el histograma para calcular algunos descriptores de las texturas, obviamente no estamos teniendo en cuenta las relaciones entre píxeles. Una posibilidad para ampliar estos descriptores es considerar no sólo la distribución de intensidades sino también la posición de píxeles con la misma o similar intensidad.

Supongamos que establecemos una dirección (horizontal, vertical, etc) y una distancia (un pixel, dos píxeles, etc) en una imagen. Entonces, el elemento (i, j) de la *matriz de co-ocurrencia* \mathbf{P} para un objeto, es el número de veces dividido por M , que los niveles i y j ocurren en dos píxeles separados por esa distancia antes definida y en la dirección dada, siendo M el número de parejas de píxeles que contribuyen a \mathbf{P} . La matriz \mathbf{P} tiene tamaño $N \times N$ donde N es el número de niveles de gris.

Claramente podemos calcular diferentes matrices de co-ocurrencia para diferentes distancias y direcciones. Sin embargo, estas matrices suelen tener bastantes ceros y para evitarlo se suele dividir el rango de los niveles de gris a alrededor de 8.

Una vez que la matriz de co-ocurrencia ha sido calculada podemos calcular diferentes descriptores de la región. Por ejemplo tenemos

1. *Entropía*

$$H = \sum_i \sum_j P_{ij} \log P_{ij}$$

2. *Inercia*

$$I = \sum_{i,j} (i - j)^2 P_{ij}$$

3. *Energía*

$$E = \sum_{i,j} [P_{ij}^2]$$

10.6. Ejercicios

1. Implementar la aproximación poligonal de una frontera.
2. Implementar de descripción de una frontera mediante Fourier
3. Calcular algunos descriptores de una región.

Capítulo 11

Introducción a la Morfología Matemática

11.1. Introducción

La palabra morfología usualmente denota la rama de la Biología que se dedica al estudio de la forma y estructura de los animales y plantas. Usaremos aquí la misma palabra en el contexto de morfología matemática.

La morfología matemática, que comenzó a finales de los años sesenta, forma, podría decirse, un cuerpo separado dentro del Análisis de Imágenes. Sus principales protagonistas son Matheron ([34]) y Serra ([47]). En este capítulo utilizaremos como referencias básicas [13] y [17].

La aproximación no morfológica al procesamiento de imágenes está próxima al cálculo, basándose como ya sabemos en los conceptos de delta de Dirac, *PSF*, y transformaciones lineales como la convolución. La morfología matemática se basa en geometría y forma, las operaciones morfológicas simplifican imágenes y conservan las principales características de formas de los objetos.

Un sistema de operadores como los de la morfología matemática es útil porque pueden formarse composiciones de sus operadores, que cuando actúan sobre formas complejas, son capaces de descomponerlas en sus partes que tienen sentido y separarlas de las partes que le son extrañas. Un sistema de operadores de este tipo y su composición permite que las formas subyacentes sean identificadas y reconstruidas de forma óptima a partir de sus formas distorsionadas y ruidosas. Además permite que cada forma se entienda en

función de una descomposición, siendo cada entidad de esa descomposición una forma simple apropiada.

Un ejemplo familiar de sistema algebraico no morfológico, como hemos dicho, es la convolución y su representación en el dominio de las frecuencias. En este contexto cualquier función de duración finita f puede interpretarse como la suma de funciones sinusoidales. La distorsión de f puede modelarse como la adición de sinusoidales o la convolución con algún núcleo. Cualquiera que sea la distorsión, el comprender que ocurre en términos de ondas permite el desarrollo de procedimientos para deshacer la distorsión no deseada, o la convolución no deseada, así como estimar, reconstruir, extraer o reconocer la imagen f original basándonos en la observación de la imagen distorsionada.

Lo que el álgebra de convolución hace con los sistemas lineales, lo hace la morfología matemática con las formas. Puesto que las formas contienen mucha información en la visión artificial, su importancia es evidente. Las operaciones morfológicas pueden simplificar los datos de la imagen, preservar las características esenciales y eliminar aspectos irrelevantes. Teniendo en cuenta que la identificación y descomposición de objetos, la extracción de rasgos, la localización de defectos e incluso los defectos en líneas de ensamblaje están sumamente relacionados con las formas, es obvio el papel de la morfología en visión artificial.

Las transformaciones morfológicas, si se usan, constituyen usualmente una parte intermedia de la secuencia del procesamiento de imágenes. En una primera fase, la imagen es digitalizada y preprocesada usando operadores de convolución locales y luego es segmentada para obtener una imagen binaria en la que se separan los objetos del fondo. Las operaciones morfológicas pueden formar una segunda fase que opera sobre la forma de esos objetos. El último paso del procesamiento evalúa los resultados de la morfología usando descriptores numéricos o sintácticos.

La morfología matemática se puede usar, entre otros, con los siguientes objetivos:

1. Pre-procesamiento de imágenes (supresión de ruido, simplificación de formas),
2. Destacar la estructura de los objetos (extraer el esqueleto, marcado de objetos, envolvente convexa, ampliación, reducción),
3. Descripción cualitativa de objetos (área, perímetro, etc).

11.2. Operaciones básicas sobre conjuntos

Sean A y B son conjuntos en un n -espacio E^n con elementos $a = (a_1, \dots, a_n)$ y $b = (b_1, \dots, b_n)$ respectivamente siendo ambos n -uplas.

1. La traslación de A por $x \in E^n$ que se nota A_x se define como

$$A_x = \{c | c = a + x, \text{ para algún } a \in A\}.$$

2. La reflexión de B notada \hat{B} se define como

$$\hat{B} = \{x | x = -b, \text{ para algún } b \in B\}.$$

3. Por último la diferencia de dos conjuntos A y B , notada $A - B$, se define mediante

$$A - B = \{x | x \in A, x \notin B\}$$

11.3. Morfología Binaria

El lenguaje de la morfología matemática binaria es el de la teoría de conjuntos. Los conjuntos en morfología matemática representan las formas presentes en imágenes binarias o de niveles de gris. El conjunto de todos los píxeles blancos en una imagen en blanco y negro (binaria) constituye una descripción completa de la imagen.

Los puntos en un conjunto sobre los que se aplica la transformación son el conjunto de puntos seleccionado y el complementario el no seleccionado. En las imágenes binarias los puntos seleccionados son los que no pertenecen al fondo.

Las operaciones primarias morfológicas son la erosión y la dilatación. A partir de ellas podemos componer las operaciones de apertura y clausura. Son estas dos operaciones las que tienen mucha relación con la representación de formas, la descomposición y la extracción de primitivas.

11.3.1. Dilatación Binaria

La dilatación es la transformación morfológica que combina dos vectores utilizando la suma. La dilatación binaria fue usada primero por Minkowski, y en la literatura matemática recibe el nombre de suma de Minkowski. Si A y B son conjuntos en un n -espacio E^n

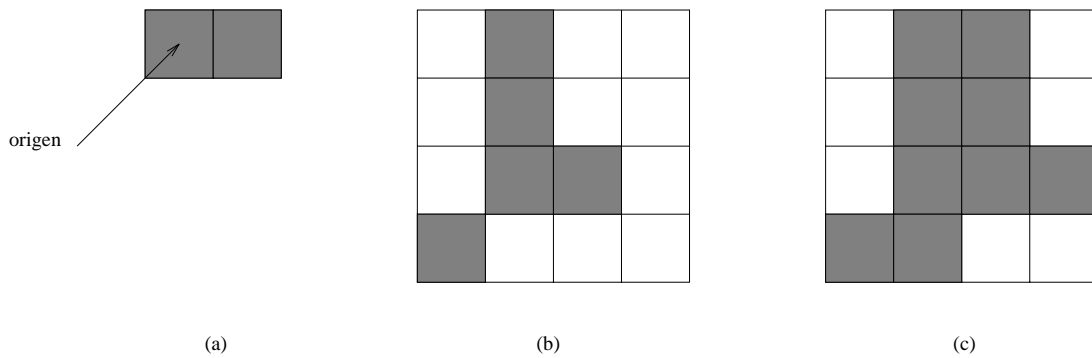


Figura 11.1: Ejemplo de dilatación. (a) Elemento estructural, B . (b) Imagen, A . (c) Resultado de la dilatación $A \oplus B$

con elementos $a = (a_1, \dots, a_n)$ y $b = (b_1, \dots, b_n)$, respectivamente, siendo ambos n -uplas, entonces la dilatación de A por B es el conjunto de todos los posibles vectores que son suma de pares de elementos, uno de A y otro de B .

Más formalmente, la dilatación de A por B se nota $A \oplus B$ y se define mediante

$$A \oplus B = \{c \in E^n \mid c = a + b \text{ para algún } a \in A \text{ y } b \in B\}$$

al ser la suma conmutativa, la dilatación también lo es: $A \oplus B = B \oplus A$.

Se puede probar que las siguientes definiciones de la dilatación son equivalentes

$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\} = \cup_{b \in B} A_b$$

En la práctica los conjuntos A y B no son simétricos. El primer elemento de la dilatación, A , está asociado con la imagen que se está procesando y el segundo recibe el nombre de elemento estructural, la forma que actúa sobre A en la dilatación para producir $A \oplus B$.

Cuando se realiza una dilatación con un elemento estructural que contiene el cero, lo que realizamos es la expansión de una imagen y es fácil pensar en una implementación paralela. Algunos ejemplos se muestran en las figuras 11.1, 11.2. Es importante tener en cuenta que el sistema de coordenadas que se usará en este tema es (fila, columna).

La dilatación tiene las siguientes propiedades

1. La dilatación por el trasladado de un elemento estructural es el trasladado de la dilatación:

$$A \oplus B_t = (A \oplus B)_t.$$

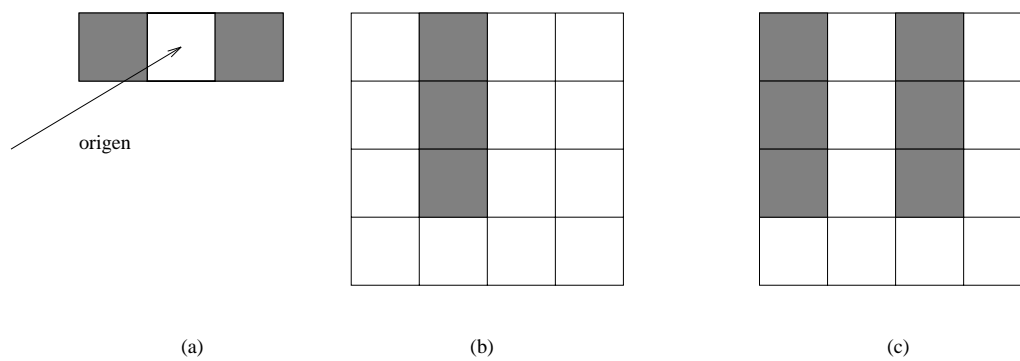


Figura 11.2: Otro ejemplo de dilatación. (a) Elemento estructural, B . (b) Imagen, A . (c) Resultado de la dilatación $A \oplus B$

2. Propiedad distributiva:

$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C).$$

3. Asociatividad (iteración):

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C.$$

4. Crecimiento:

$$A \subseteq B \Rightarrow A \oplus K \subseteq B \oplus K \quad \forall K.$$

11.3.2. Erosión binaria

La erosión es la operación morfológica dual, un concepto que definiremos formalmente a continuación, de la dilatación. Es la transformación morfológica que combina dos conjuntos usando el concepto de inclusión. Si A y B son conjuntos en el espacio euclídeo n -dimensional, entonces la erosión de A por B es el conjunto de todos los elementos x para los que $x + b \in A$ para todo $b \in B$. La sustracción de Minkowski está muy relacionada con la erosión.

La erosión de A por B se nota $A \ominus B$ y su definición es

$$A \ominus B = \{x \in E^n \mid x + b \in A \text{ para todo } b \in B\}$$

un ejemplo de erosión se encuentra en la figura 11.3.

La utilidad de la erosión puede apreciarse mejor cuando ésta se expresa de forma diferente. La erosión de una imagen, A , por un elemento estructural, B , es el conjunto de

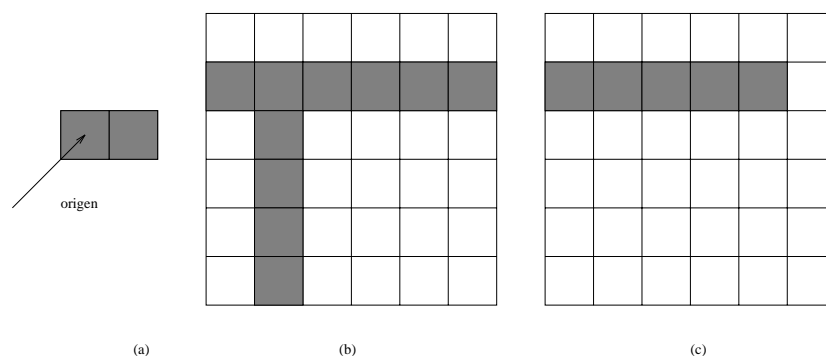


Figura 11.3: Ejemplo de erosión. (a) Elemento estructural, B . (b) Imagen, A . (c) Resultado de la erosión $A \ominus B$

todos los elementos $x \in E^n$ para los cuales B trasladado por x está contenido en A . La demostración es inmediata y se tiene

$$A \ominus B = \{x \in E^n \mid B_x \subseteq A\}$$

Mientras que la dilatación puede representarse como la unión de los trasladados, la erosión puede representarse como la intersección de los trasladados negativos.

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

La erosión se concibe usualmente como una reducción de la imagen original. En términos de teoría de conjuntos, el conjunto erosionado se suele pensar que siempre está contenido en el original. Una transformación que cumple esta propiedad se dice antiextensiva. Sin embargo, la erosión es antiextensiva cuando el origen pertenece al elemento estructural. Esto es, si $0 \in B$, entonces $A \ominus B \subseteq A$, la demostración de esta propiedad, al igual que de las anteriores, es obvia.

Al igual que la dilatación, la erosión es también creciente: si $A \subseteq B$ entonces $A \ominus K \subseteq B \ominus K$. Además la erosión por un elemento estructural mayor produce un resultado menor si $K \subseteq L$ entonces $A \ominus L \subseteq A \ominus K$.

La dilatación y la erosión son muy similares en el sentido de que lo que uno hace al objeto el otro lo hace al fondo. Esta relación puede formularse como una relación de dualidad.

Dos operadores son duales cuando la negación de una formulación empleado en el primero es igual a la misma formulación empleando el segundo operador en la variable

negada. Un ejemplo es la ley de DeMorgan que establece la dualidad entre la unión y la intersección $(A \cup B)^c = A^c \cap B^c$, siendo la negación el complementario.

En morfología la negación de un conjunto puede ocurrir de dos formas diferentes: en un sentido lógico, siendo la negación la complementación o en un sentido geométrico, siendo la negación el cambio de la orientación de un conjunto con respecto a los ejes de coordenadas. Esta negación recibe el nombre de reflexión.

Sea $B \subseteq E^n$, como ya sabemos, la reflexión de B se nota \hat{B} y se define mediante

$$\hat{B} = \{x \mid \text{para algún } b \in B, x = -b\}$$

Como se ve en el siguiente teorema, el complemento de la erosión es la dilatación de un complemento por reflexión. La dualidad de dilatación y erosión usa negaciones lógicas y geométricas por los diferentes papeles jugados por la imagen y el elemento estructural.

Teorema 1 *Dualidad de la erosión y la dilatación.*

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

la demostración puede encontrarse en [17]

Como corolario tenemos

Corolario 1

$$(A \oplus B)^c = A^c \ominus \hat{B}$$

Algunas de las propiedades de la erosión se resumen en la lista siguiente

1. Propiedad Distributiva:

$$A \ominus (K \cup L) = (A \ominus K) \cap (A \ominus L)$$

2. Localización:

$$(A \cap B) \ominus K = (A \ominus K) \cap (B \cap K)$$

Finalmente, con respecto a la descomposición de elementos estructurales, una regla de la cadena para la erosión se verifica cuando el elemento estructural se puede descomponer mediante dilatación.

$$A \ominus (B \oplus C) = (A \ominus B) \ominus C$$

11.3.3. Algoritmos morfológicos basados en la erosión y la dilatación

Basándonos en la discusión anterior, podemos ahora abordar algunos usos prácticos de la morfología. Cuando trabajamos con imágenes binarias, la principal aplicación de la morfología es la extracción de componentes de la imagen que son útiles para la descripción y representación de formas, en particular consideraremos los algoritmos morfológicos para extraer fronteras, componentes conexas, envolventes convexas y el esqueleto de una región. También es posible elaborar varios métodos para rellenar, aumentar, disminuir y podar regiones que se usan en conjunción con estos algoritmos como pre- o post- procesamiento.

En este apartado veremos sólo los algoritmos que están basados en erosiones y dilataciones simples

Extracción de fronteras

La frontera de un conjunto A , que se nota $\beta(A)$, se puede obtener primero erosionando A por B y realizando posteriormente la diferencia entre A y su erosión. Es decir,

$$\beta(A) = A - (A \ominus B)$$

La figura 11.4 ilustra la mecánica de la extracción de fronteras. Con un elemento estructural simple. Aunque el elemento estructural usado es muy simple, existen elecciones más complejas, por ejemplo de tamaño 5×5 que ampliaría el grosor de la frontera a dos o tres píxeles

Relleno de Regiones

A continuación desarrollaremos un algoritmo simple basado en un conjunto de dilataciones, complementaciones e intersecciones para rellenar regiones. Veamos como hacerlo con un ejemplo, en la figura 11.5 tenemos una frontera, A , 8-conectada de una región, cuyo objetivo es rellenarla. Comenzamos con un punto p dentro del objeto, es decir en la región, el objetivo es rellenar la región entera con unos.

Con la hipótesis que tenemos todos los puntos no frontera son cero y determinan el complementario de A , p tiene el valor uno al comenzar el procedimiento. El siguiente procedimiento rellena la región con unos (píxeles grises).

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3 \dots$$

donde $X_0 = p$ y B es el elemento estructural que se muestra en la figura 11.5(c). El algoritmo termina en la iteración k si $X_{k-1} = X_k$. La unión de los X_k y A es la frontera y la región rellena.

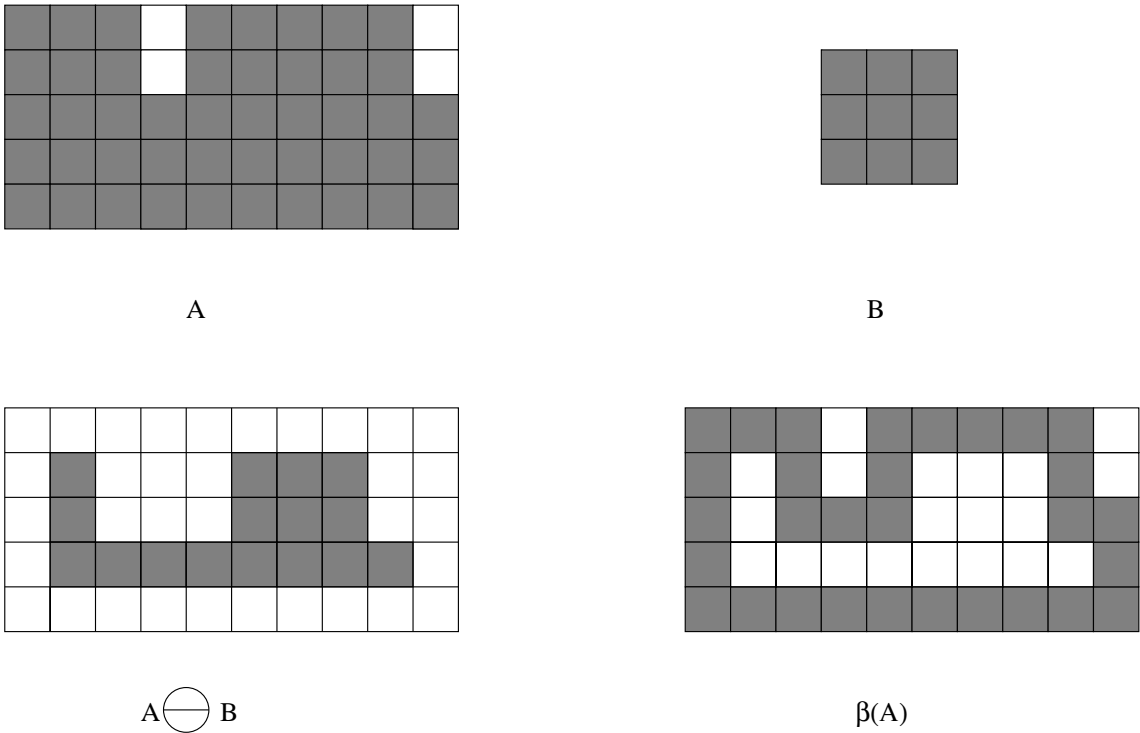


Figura 11.4: Extracción de fronteras usando la erosión

Es importante notar que el proceso anterior rellenaría la región entera si no hiciéramos la intersección con A^c , sin embargo esta intersección limita el proceso a la región de interés (este tipo de dilatación recibe a veces el nombre de *dilatación condicionada*). Es obvio que el proceso se puede extender a cualquier número de subconjuntos siempre que conozcamos puntos dentro de ellos.

Extracción de componentes conexas

Con el concepto de conectividad conocido de capítulos anteriores veamos como utilizar la morfología matemática para extraer componentes conexas, un concepto que es básico en muchas aplicaciones del análisis de imágenes. Supongamos que Y representa una componente conexa contenida en un conjunto A y supongamos que conocemos un punto p que pertenece a dicha región. Entonces, el siguiente procedimiento puede utilizarse para extraer Y

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, \dots$$

donde $X_0 = p$, B es un elemento estructural apropiado, la condición de convergencia se establece como $X_{k-1} = X_k$ con $Y = X_k$. La figura 11.6 muestra un ejemplo de aplicación, la forma del elemento estructural supone 8-conectividad en los píxeles.

11.3.4. Transformación Ganancia-Pérdida

La transformación morfológica de ganancia-pérdida es una herramienta básica para la detección de formas.

Sea $B = (J, K)$ donde J es el conjunto formado por los elementos de K asociados con el objeto y B_2 el asociado con el fondo, la **transformación ganancia-pérdida** se define como

$$A \otimes B = (A \ominus J) \cap (A^c \ominus K)$$

Por tanto, el conjunto $A \otimes B$ contiene todos los puntos en los que simultáneamente J tiene una ganancia en A y K la tiene en A^c .

Utilizando la definición de diferencia de conjuntos y la relación dual entre la erosión y la dilatación, podemos escribir la ecuación anterior como

$$A \otimes B = (A \ominus J) - (A \oplus \hat{K})$$

Un ejemplo de aplicación de la transformación ganancia-pérdida es la búsqueda de puntos aislados, si $J = \{(0, 0)\}$ y $K = \{(0, 1), (0, -1), (1, 0), (-1, 0)\}$, entonces $I \otimes B$ es el

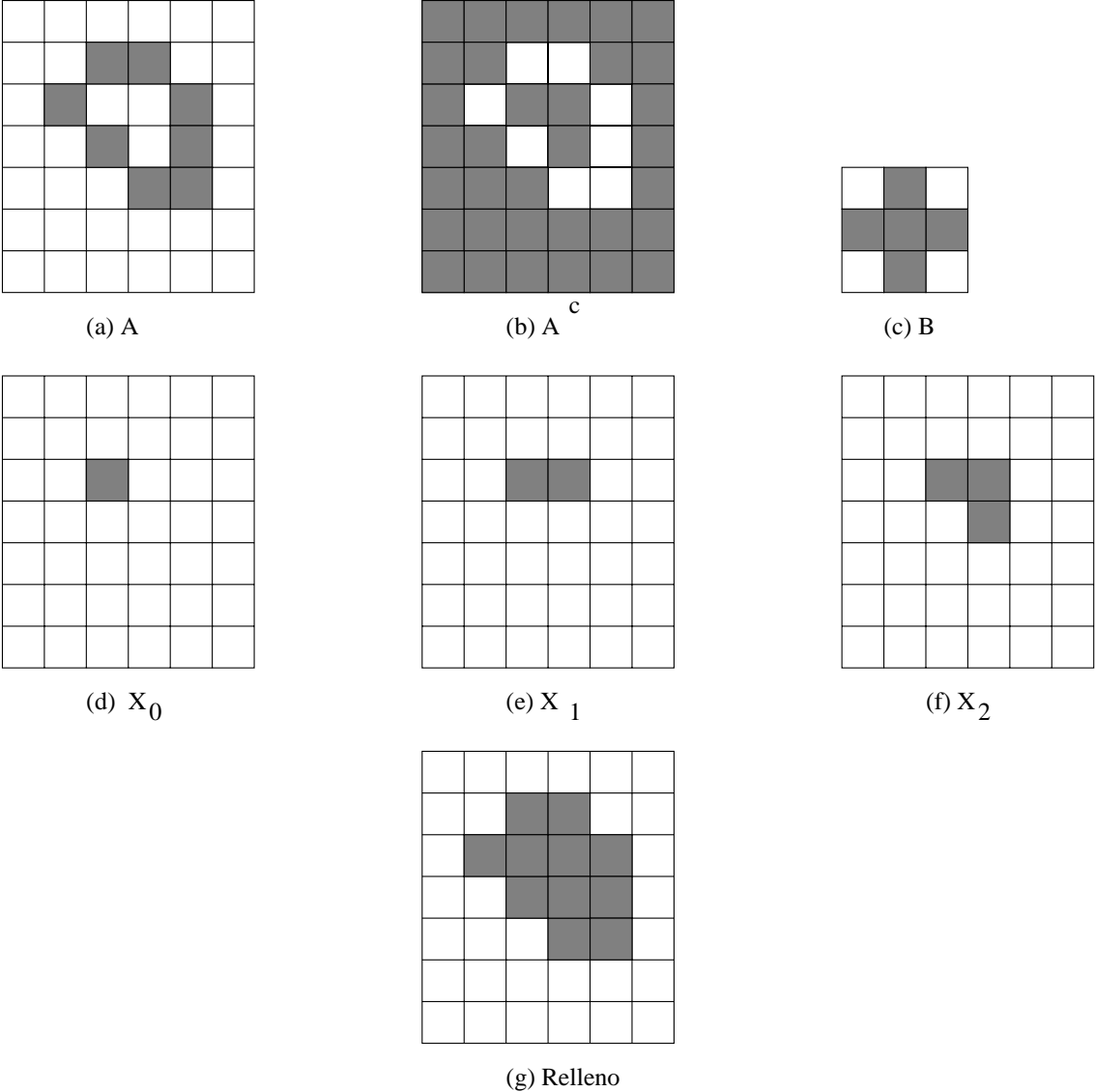


Figura 11.5: Relleno de regiones

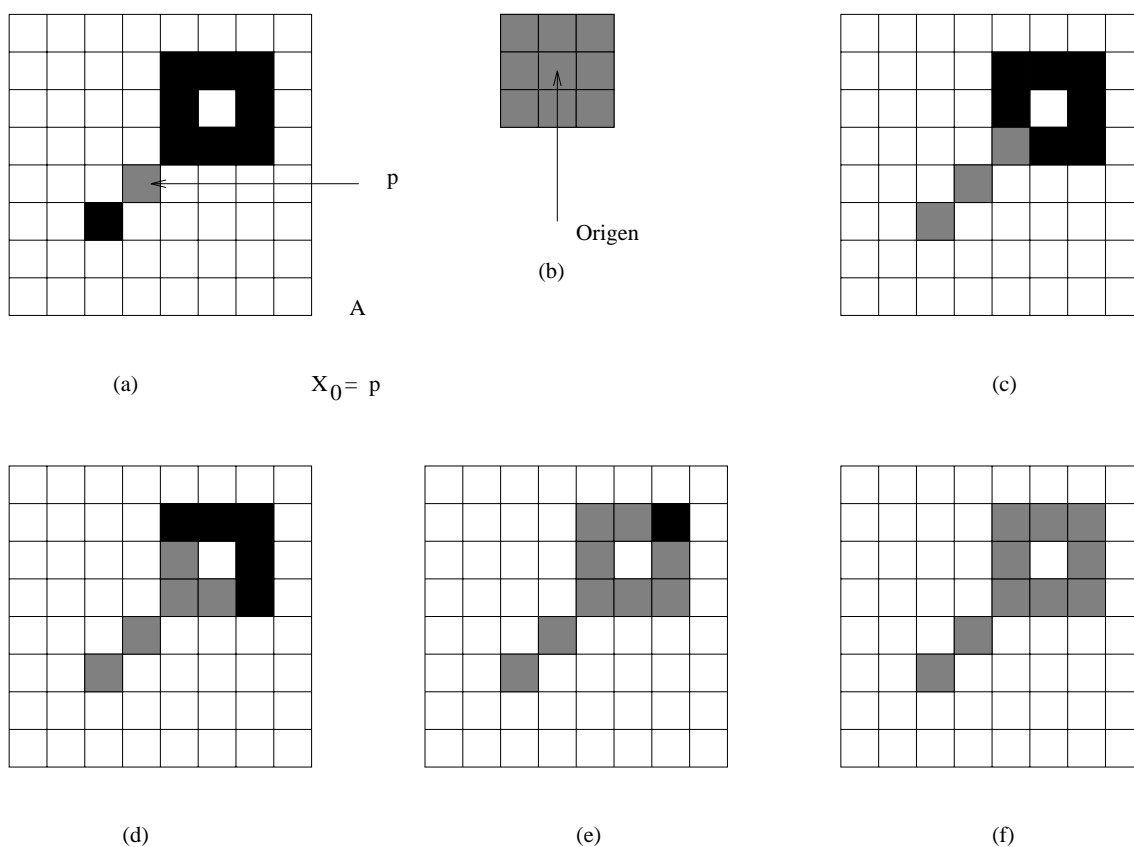


Figura 11.6: A contiene una componente conexa que comienza en p , (tono gris para distinguirlo de los otros). En (c) el resultado de la primera aplicación del proceso iterativo, (d) el segundo paso (e) el tercero y (f) el último

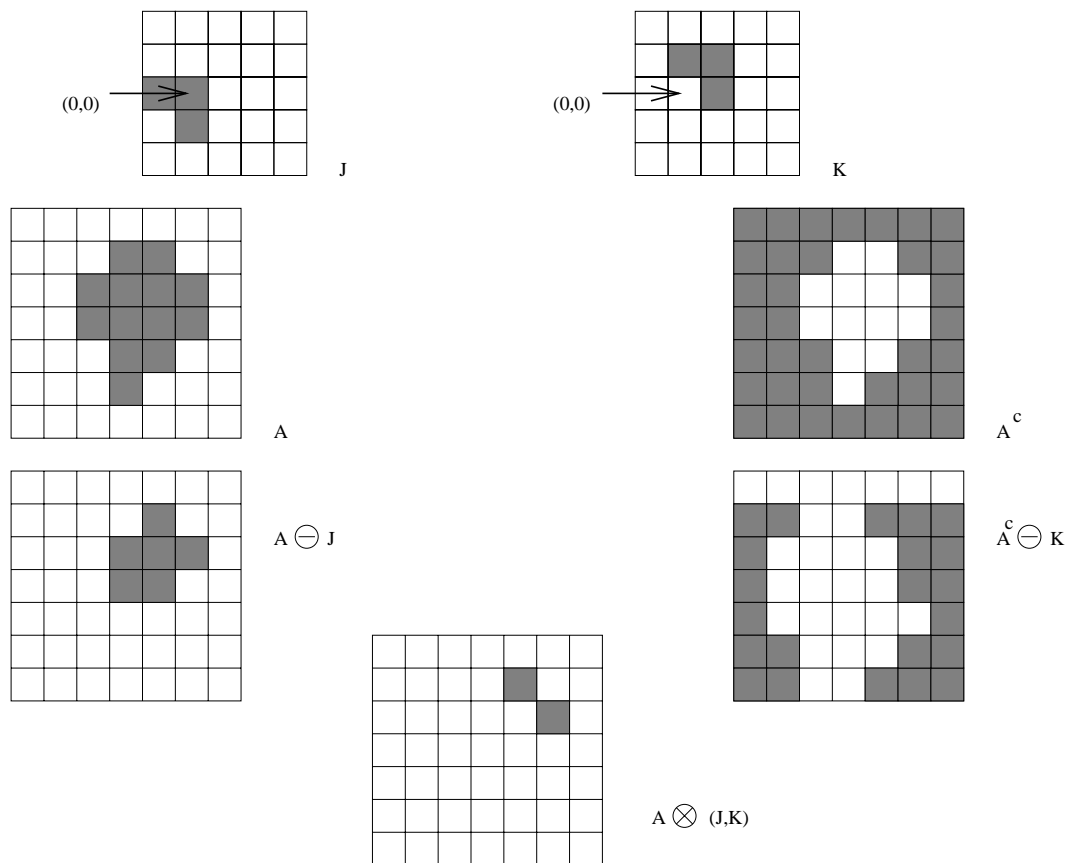


Figura 11.7: Aplicación de la transformación ganancia-pérdida para encontrar esquinas superiores derechas

conjunto de todos los píxeles 4-aislados. Por último, la figura 11.7 muestra como encontrar esquinas superiores derechas de imágenes usando transformaciones de ganancia-pérdida.

11.3.5. Algoritmos de morfología basados en la transformación Ganancia-Pérdida

Envolvente Convexa

El concepto de envolvente convexa fue ya introducido, siendo además un descriptor útil de las regiones de una imagen. Veamos ahora como obtener la envolvente convexa basándonos en la morfología matemática. Denotemos $C(A)$ la envolvente convexa de un conjunto A , sean $B^i, i = 1, 2, 3, 4$ cuatro elementos estructurales. El procedimiento consiste

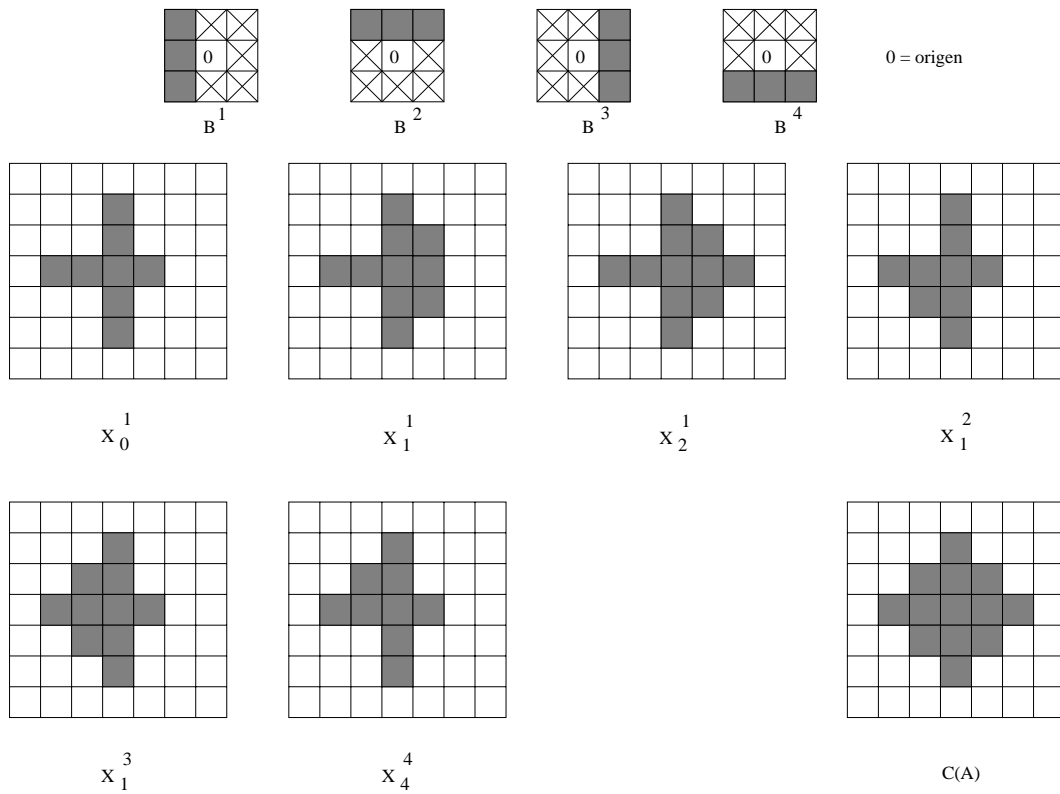


Figura 11.8: Cálculo de la envolvente convexa de un conjunto

en implementar la siguiente ecuación

$$X_k^i = (X \otimes B^i) \cup A \quad i = 1, 2, 3, 4 \text{ y } k = 1, 2, \dots$$

con $X_0^i = A$. Observemos que hemos usado la definición simplificada de la transformación ganancia-pérdida con $B^i = (B^i, W - B^i)$. Sea ahora $D^i = X_{\text{CONV}}^i$, donde el subíndice conv indica convergencia en el sentido de que $X_k = X_{k-1}$. Entonces, la envolvente convexa de A es

$$C(A) = \cup_{i=1}^4 D^i$$

La figura 11.8 ilustra con un ejemplo el procedimiento para extraer la envolvente convexa. El origen está siempre en el centro del elemento estructural y los píxeles marcados \times no se usan para contruir el complementario del elemento estructural

Disminución

La disminución de un conjunto A por un elemento estructural B puede ser definida

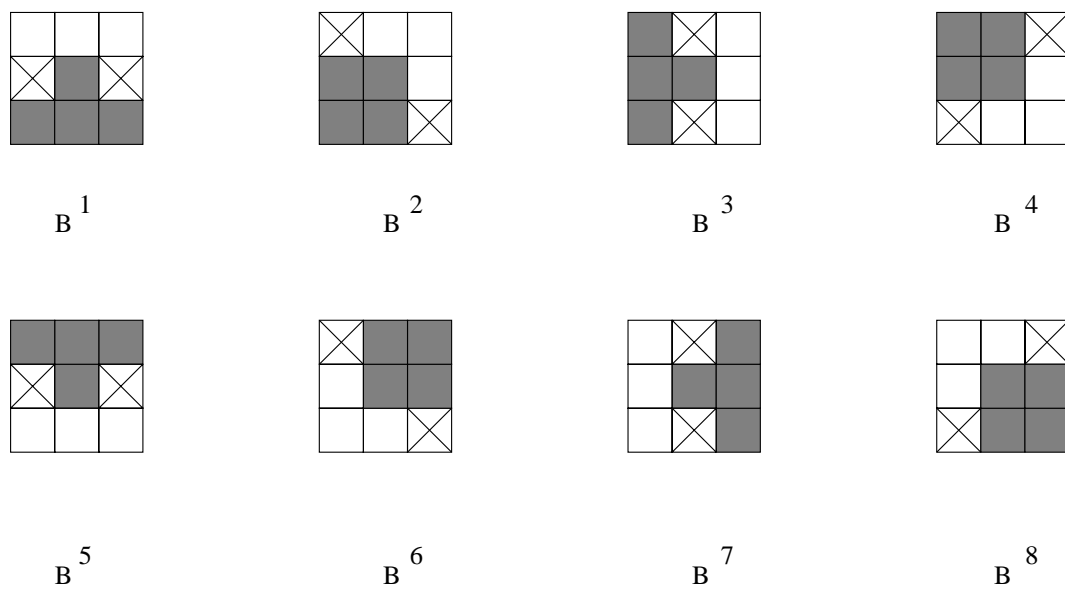


Figura 11.9: Elementos estructurales para el cálculo de la m-conectividad

en términos de la transformación ganancia-pérdida

$$A \oslash B = A - (A \otimes B) = A \cup (A \otimes B)^c$$

Una definición más útil para el adelgazamiento de A simétrico está basado en una sucesión de elementos estructurales como

$$\{B\} = \{B^1, B^2, \dots, B^n\}$$

donde B^i es una versión rotada de B^{i-1} . Usando este concepto definimos el adelgazamiento por una sucesión de elementos estructurales como

$$A \oslash \{B\} = ((\dots((A \oslash B^1) \oslash B^2) \dots) \oslash B^n)$$

en otras palabras el proceso es adelgazar A un paso con B^1 , entonces adelgazar el resultado con B^2 y así sucesivamente hasta B^n . El proceso se repite hasta que no ocurren cambios.

La figura 11.9 muestra los elementos estructurales que son necesarios para construir la m-conectividad de una región.

Ampliación

La ampliación es el dual morfológico de la disminución y se define mediante la expresión

$$A \odot B = A \cup (A \otimes B)$$

donde B es un elemento estructural apropiado para la ampliación. Al igual que el adelgazamiento, la ampliación se puede definir también secuencialmente teniendo

$$A \odot \{B\} = ((\dots((A \odot B^1) \odot B^2) \dots) \odot B^n)$$

en el caso de la ampliación los elementos estructurales que se usan cambian los ceros por unos. Sin embargo, esta implementación directa no se suele usar, lo que se hace es adelgazar el fondo y luego calcular el complementario. En otras palabras, para ampliar A , calculamos $C = A^c$, lo adelgazamos creando C y luego calculamos C^c .

11.3.6. Apertura y Clausura

Con los conceptos que hemos visto podemos comprender ahora mejor las razones por las que la dilatación y la erosión están muy relacionados con la forma. La erosión y la dilatación usualmente se emplean por pares, bien la dilatación seguida por la erosión o al revés. En cualquier caso, el resultado de esta aplicación sucesiva de erosiones y dilataciones es una eliminación de detalles menores que el elemento estructural, sin distorsionar la forma global del objeto.

Un aspecto fundamental de esta aplicación sucesiva es el hecho de que la aplicación sucesiva de erosiones y dilataciones es idempotente, es decir la aplicación sucesiva de estos operadores no cambia el resultado. La importancia práctica de este resultado es que las transformaciones idempotentes forman un estado completo de los algoritmos del análisis de imágenes ya que las formas podrían ser descritas en términos de los elementos estructurales bajo los cuales pueden ser abiertos o cerrados y quedarse igual.

Podría decirse que la apertura y la clausura corresponden dentro de la morfología a lo que las proyecciones ortogonales son al álgebra lineal. Una proyección ortogonal es idempotente y selecciona la parte de un vector que está en un determinado subespacio. Análogamente, la apertura y la clausura proporcionan los medios por los cuales seleccionar subformas y superformas de una forma compleja.

Como hemos visto, cuando el elemento estructural contiene el origen, la dilatación expande la imagen y la erosión la reduce. En esta sección discutiremos otras dos importantes operaciones morfológicas: apertura y clausura. La apertura generalmente suaviza los contornos de una imagen.

La apertura de A por un elemento estructural K se nota $A \circ K$, se define como

$$A \circ K = (A \ominus K) \oplus K$$

que en palabras establece que la apertura de A por K es simplemente la erosión de A por K , seguido de la dilatación del resultado por K .

La clausura de A por un elemento estructural K se nota $A \bullet K$, se define como

$$A \bullet K = (A \oplus K) \ominus K$$

que en palabras establece que la clausura de A por K es simplemente la dilatación de A por K , seguido de la erosión del resultado por K .

Si A no cambia con la apertura con K diremos que A es abierto con respecto a K . Si A no cambia con la clausura con K diremos que A es cerrado con respecto a K .

La habilidad de una apertura para seleccionar de un conjunto el subconjunto que se acopla al elemento estructural es clara por el teorema de caracterización

$$A \circ K = \{x \in A \mid \text{para algún } t \in A \ominus K, x \in K_t \text{ y } K_t \subseteq A\}$$

La apertura de A por K selecciona los puntos de A que se acoplan a K en el sentido de que los puntos pueden ser cubiertos por alguna traslación del elemento estructural K que esté contenido completamente en A . Si tomamos un disco como elemento estructural suavizaremos los contornos, romperemos uniones estrechas entre partes de conjuntos y eliminaremos salientes estrechos. La clausura tiende a suavizar las secciones de contornos pero en sentido inverso, une separaciones estrechas y golfos (formas \supset) estrechos, elimina huecos y pequeñas discontinuidades en las fronteras.

Reorganizando la información del teorema de caracterización tendremos

$$A \circ K = \bigcup_{y \in A \ominus K} K_y = \bigcup_{K_y \subseteq A} K_y$$

lo que produce una imagen muy clara de lo que hace la apertura. La apertura $A \circ K$ se obtiene pasando el elemento estructural K dentro de A y no permitiéndole que salga. Hemos probado algo más, de la fórmula anterior se deduce que $A \circ K = A \circ K_x$ para cualquier x .

Veamos una interpretación geométrica simple de la apertura. Supongamos que consideramos el disco que constituye un elemento estructural K como una bola plana que rueda en R^2 . La frontera de $A \circ K$ viene dada por los puntos de la frontera de K que alcanzan la parte más lejana de la frontera de A cuando K rueda alrededor de la parte interior de la frontera.

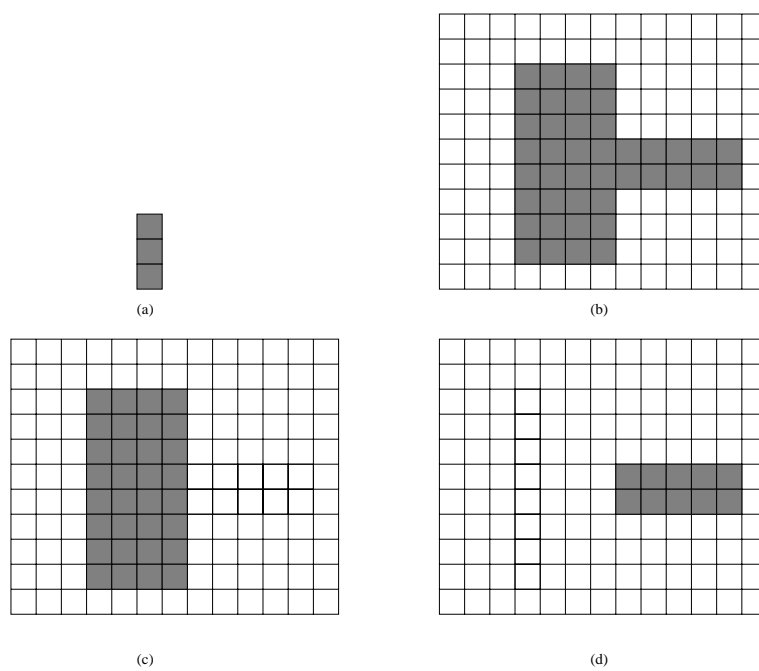


Figura 11.10: Extracción de cuerpo y asa. (a) contiene la imagen A , (b) el elemento estructural K , (c) la apertura $A \circ K$ y (d) $A - A \circ K$.

La figura 11.10 ilustra como podemos usar la apertura para descomponer objetos. Supongamos un cuadrado unido a un asa. El procedimiento descrito en la figura nos sirve para separar las dos partes.

Se cumple la dualidad entre apertura y clausura, es decir,

$$(A \bullet K)^c = A^c \circ \hat{K}$$

El teorema de caracterización para la apertura y la dualidad entre apertura y clausura nos lleva a la caracterización de la clausura que establece que

$$A \bullet K = \{x | x \in \hat{K}_t \text{ implica } \hat{K}_t \cap A \neq \emptyset\}$$

La clausura de A incluye todos los puntos que cumplen la condición de que cuando son cubiertos por un trasladado del reflejado del elemento estructural, este trasladado y A deben tener intersección no vacía. De nuevo esta transformación es invariante por traslaciones del elemento estructural.

La apertura y clausura con un elemento estructural isotrópico se usa para eliminar detalles específicos más pequeños que el elemento estructural sin distorsionar la forma global de los objetos.

De los teoremas de caracterización se sigue que la apertura es antiextensiva, $A \circ K \subseteq A$, y la clausura es extensiva, $A \subseteq A \bullet K$.

Otro aspecto significativo de estas operaciones es que son idempotentes, es decir,

$$\begin{aligned} X \circ B &= (X \circ B) \circ B \\ X \bullet B &= (X \bullet B) \bullet B \end{aligned}$$

11.3.7. Algoritmos morfológicos basados en aperturas y clausuras

El concepto de esqueleto ya fue introducido en capítulos anteriores. Veamos como se puede calcular morfológicamente.

Serra ([47]) entre otros probó que el esqueleto de un conjunto A puede ser expresado en términos de erosiones y aperturas. Esto es, con $S(A)$ notando el esqueleto de A , se puede probar que

$$S(A) = \cup_{l=0}^L S_l(A)$$

con

$$S_k(A) = \cup_{l=0}^k \{(A \ominus lK) - [(A \ominus lK) \circ K]\}$$

donde $A \ominus lK$ denota la aplicación sucesiva de l erosiones a A y L es el último paso iterativo antes de que A se erosione a un conjunto vacío. En otras palabras,

$$L = \text{máx}\{l | (A \ominus lK) \neq \emptyset\}$$

La formulación dada en las ecuaciones anteriores establece que el esqueleto de A puede obtenerse como la unión de los subconjuntos de esqueletos $S_k(A)$. Puede probarse también que

$$A = \cup_{l=0}^L (S_l(A) \oplus lK)$$

11.4. Morfología de niveles de gris

En esta sección extenderemos a imágenes de niveles de gris, las operaciones básicas de dilatación, erosión, apertura y clausura y usaremos estos operadores para el desarrollo de algunos algoritmos básicos. De nuevo, el objetivo es el uso de la morfología de niveles de gris para extraer componentes de la imagen que son útiles en la representación y descripción de las formas. En particular veremos algoritmos para extraer fronteras de imágenes mediante el gradiente morfológico y para la partición de regiones basada en su contenido en textura. También discutiremos algoritmos para alisamiento y realce que de nuevo pueden ser pasos en pre- post- procesamiento.

Vamos, por simplicidad, a formular el estudio en Z^2 aunque es posible su realización en otros espacios. Trabajaremos con imágenes de la forma $f(x, y)$ y $b(x, y)$ donde $f(x, y)$ es una imagen de entrada y $b(x, y)$ un elemento estructural, de hecho una subimagen. Supondremos que las imágenes son discretas, es decir si Z denota los enteros, supondremos que (x, y) son enteros de $Z \times Z$ y que f y b son funciones que asignan un número real a cada par de coordenadas (x, y) , si los niveles de gris son enteros, Z sustituirá a R .

11.4.1. Dilatación

La dilatación de una imagen de niveles de gris f por b que se notará $f \oplus b$, se define como

$$(f \oplus b)(s, t) = \text{máx}\{f(s - x, t - y) + b(x, y) | (s - x, t - y) \in D_f, (x, y) \in D_b\}$$

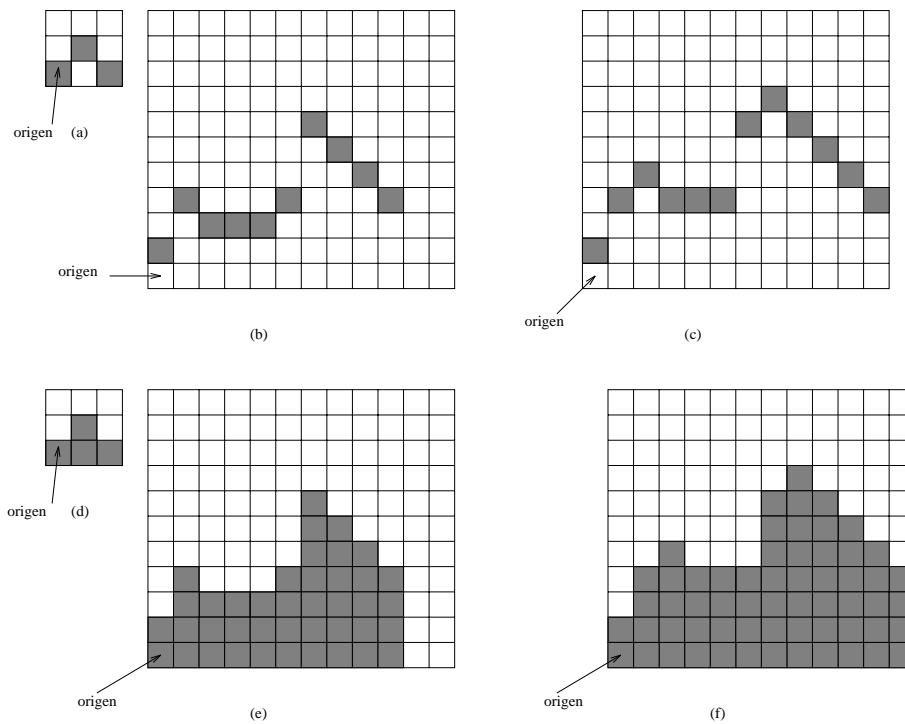


Figura 11.11: (a) Elemento estructural b , (b) imagen f , (c) $f \oplus b$. (d) relleno de b , notado $U(b)$, (e) relleno de f , notado $U(f)$, la extracción de la parte superior de $U(f) \oplus U(b)$ que se nota $T(U(f) \oplus U(k))$ produce $f \oplus b$.

donde D_f y D_b son los dominios de f y b respectivamente. Como antes b es el elemento estructural, pero ahora es una función.

Observemos que se impone la condición de inclusión en el espacio de definición y que la ecuación anterior es casi como una convolución con el máximo reemplazando a la suma de la convolución y la suma al producto con el nucleo de convolución..

En la figura 11.11 se ilustra un ejemplo de dilatación para niveles de grises. Puede probarse que el efecto de la dilatación es como si tuvieramos una imagen que fuese el relleno de la función y un elemento estructural que fuese el relleno también de la función elemento estructural y calculásemos la dilatación binaria clásica extrayendo luego sólo la parte superior de la imagen.

11.4.2. Erosión

La erosión de niveles de grises, que se nota $f \ominus b$, se define mediante

$$(f \ominus b)(s, t) = \min\{f(s + x, t + y) - b(x, y) \mid (s + x, t + y) \in D_f, (x, y) \in D_b\}$$

donde, de nuevo, D_f y D_b son los dominios de f y b respectivamente.

Observemos que se impone la condición de inclusión en el espacio de definición y que la ecuación anterior es casi como una correlación con el máximo reemplazando la suma y la diferencia el producto.

En la figura 11.12 se ilustra un ejemplo de erosión para niveles de grises. Puede probarse que el efecto de la erosión es como si tuvieramos una imagen que fuese el relleno de la función y un elemento estructural que fuese el relleno también de la función elemento estructural y calculásemos la erosión binaria clásica extrayendo luego sólo la parte superior de la imagen.

Al igual que antes la erosión y la dilatación, ahora para niveles de grises son operaciones duales con respecto a la complementación y reflexión, es decir,

$$(f \ominus b)^c(x, y) = (f^c \oplus \hat{b})(x, y)$$

donde $f^c(x, y) = -f(x, y)$ y $\hat{b} = b(-x, -y)$.

11.4.3. Apertura y Clausura

Las expresiones para la apertura y clausura de niveles de gris tienen la misma forma que la apertura y clausura binarias. La apertura de f por un elemento estructural b , se nota $f \circ b$ y se define mediante

$$f \circ b = (f \ominus b) \oplus b$$

Análogamente, la clausura de f por b se nota $f \bullet b$, y se define

$$f \bullet b = (f \oplus b) \ominus b$$

La apertura y la clausura son duales con respecto a la complementación y reflexión. Es decir,

$$(f \bullet b)^c = f^c \circ \hat{b}$$

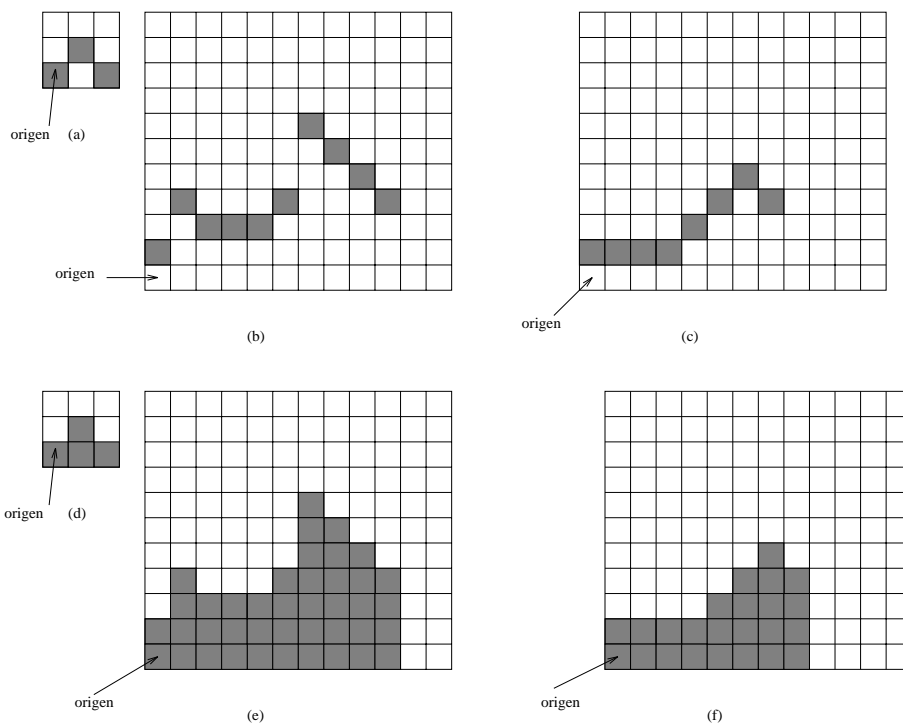


Figura 11.12: (a) Elemento estructural b , (b) imagen f , (c) $f \ominus b$. (d) relleno de b , notado $U(b)$, (e) relleno de f , notado $U(f)$, la extracción de la parte superior de $U(f) \ominus U(b)$ que se nota $T(U(f) \ominus U(k))$ produce $f \ominus b$.

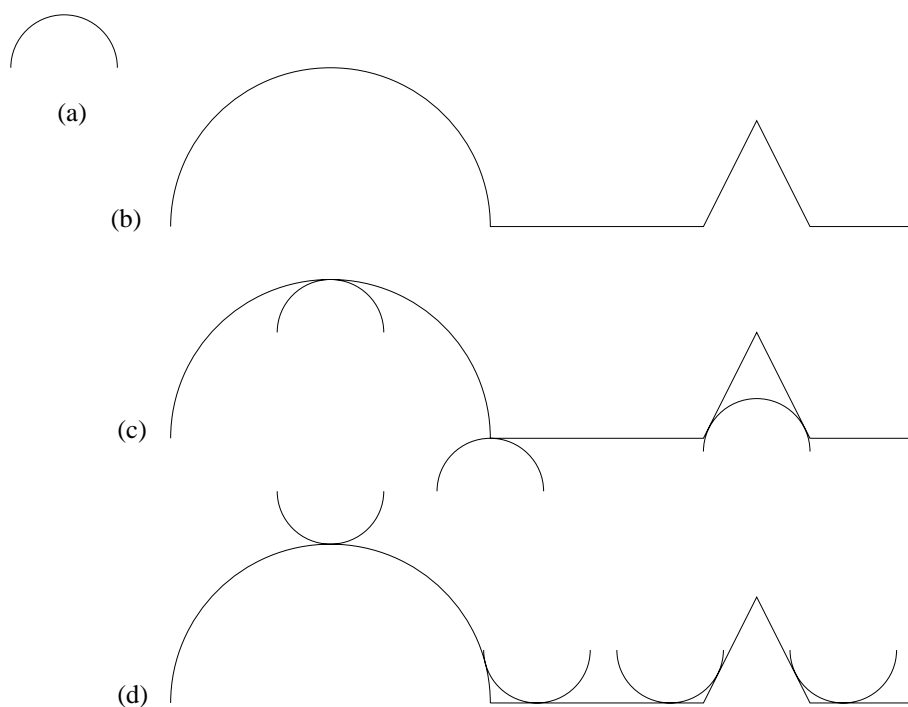


Figura 11.13: (a) Elemento estructural que se interpreta como esfera, (b) imagen, (c) proceso para obtener la apertura, (d) proceso para obtener la clausura

La apertura y la clausura de una imagen tienen una interpretación geométrica simple. Supongamos que vemos una función de imagen $f(x, y)$ en perspectiva 3 - D siendo x e y las coordenadas usuales y la tercera el nivel de brillo (la función). En esta representación, la imagen aparece como una superficie discreta cuyo valor en cualquier punto (x, y) es el valor de la imagen. Supongamos que queremos realizar la apertura por un elemento esférico b (de hecho es sólo la semiesfera lo que necesitamos) y vemos el elemento estructural como una bola que rueda. Entonces el mecanismo para abrir f por b puede interpretarse geoméricamente como el proceso de hacer rodar la bola por la parte inferior de la función y la clausura conlleva hacerla rodar por la parte superior. La figura 11.13 ilustra el procedimiento.

11.4.4. Algunas aplicaciones de la morfología de niveles de gris

Terminaremos este estudio sobre la morfología matemática analizando algunas aplicaciones de la misma.

Alisamiento morfológico

Una forma de llevar a cabo el alisamiento de una imagen es realizar una apertura seguida de una clausura. El resultado de estas dos operaciones es suprimir o atenuar elementos extraños muy brillantes u oscuros.

Gradiente morfológico

Además de las operaciones antes discutidas para la supresión de elementos extraños, a menudo la dilatación y la erosión se usan para calcular el gradiente morfológico de una imagen que se nota g y se define

$$g = f \oplus b - f \ominus b$$

Transformación Top-Hat

La transformación top-hat notada h se define como

$$h = f - f \circ b$$

donde como antes f es la imagen y b el elemento estructural. El nombre se debe a la forma del elemento estructural que se usa. Esta transformación es útil para resaltar detalles en la presencia de sombras.

Ejercicios

1. Implementar la erosión y dilatación binaria.
2. Implementar la apertura y clausura binarias.
3. Calcular las componentes conexas de un conjunto.
4. Implementar la erosión y dilatación para niveles de gris.

Capítulo 12

Introducción a la Compresión de Imágenes

12.1. Introducción

La compresión de imágenes trata de minimizar el número de bits necesarios para representar una imagen. Las aplicaciones de la compresión de imágenes son principalmente la transmisión y almacenamiento de información. En transmisión, sus principales aplicaciones son televisión, *remote sensing*, radar y sonar, teleconferencias, comunicación por computadora, transmisión por fax y otras así. En almacenamiento, la compresión de imágenes se usa sobre documentos, imágenes médicas, secuencias de imágenes, imágenes de satélite, mapas meteorológicos, etc. También es posible crear algoritmos rápidos que trabajen con las imágenes comprimidas con lo que el número de operaciones requeridas para implementar los algoritmos se reduce.

Teniendo en cuenta que una imagen de televisión tiene aproximadamente 512×512 pixels, usando 8 bits por pixel para cada canal de color y 30 imágenes por segundo, nos obliga a disponer de un canal y memoria suficiente para transmitir 180×10^6 bits por segundo. Dependiendo de las aplicaciones, la cantidad de datos para imágenes digitales sin procesar puede variar entre 10^5 y 10^8 bits por imagen. Esto hace que las técnicas de compresión de imágenes sean deseables.

12.1.1. Ratios de Información

La cantidad de datos en imágenes sin procesar no representa necesariamente su cantidad de información media que, para una fuente con L símbolos independientes posibles con probabilidades p_i , $i = 0, \dots, L - 1$, viene dada por la entropía

$$H = - \sum_{i=0}^{L-1} p_i \log_2 p_i \text{ bits por símbolo} \quad (12.1)$$

Según el *teorema de codificación sin ruido de Shannon* es posible codificar, sin distorsión, una fuente de entropía H bits por símbolo usando $H + \varepsilon$ bits por símbolo, donde ε es una cantidad positiva arbitrariamente pequeña. Así pues, la máxima compresión posible C , definida por

$$C = \frac{\text{número medio de bits de los datos sin procesar } (B)}{\text{número medio de bits de los datos codificados } (H + \varepsilon)} \quad (12.2)$$

es $B/(H + \varepsilon) \simeq B/H$. Sin embargo, la computación de este ratio de compresión para imágenes es impráctica, si no imposible. Por ejemplo, una imagen digital $N \times M$ con B bits por pixel es uno de los $L = 2^{BNM}$ patrones posibles que podrían aparecer. Si se conociera p_i , la probabilidad del i -ésimo patrón, se podría calcular la cantidad de información para imágenes $N \times M$ con B bits por pixel. Entonces se podrían almacenar todos los L posibles patrones de imágenes y codificar la imagen por su dirección – usando un método de codificación apropiado, que requerirá aproximadamente H bits por imagen o, bien, H/NM bits por pixel.

Este tipo de codificación se llama *cuantificación de vector (vector quantization)* o *codificación por bloques (block coding)*. La principal dificultad de este método es que, incluso con valores pequeños de N y M , L puede ser prohibitivamente grande; por ejemplo, para $B = 8$, $N = M = 16$, resulta $L = 2^{2048} \simeq 10^{614}$. Una idea práctica es usar bloques pequeños, digamos 4×4 , normalizar cada bloque para que tenga media 0 y varianza 1 y usando unas pocas imágenes tipo para entrenamiento, almacenar los $L' \ll L$ patrones con mayor probabilidad. Si el bloque de entrada es uno de esos L' patrones, codificarlo por la dirección del bloque; si no reemplazarlo por su valor medio.

12.2. Cuantificación

Un cuantificador hace la correspondencia de una variable continua u en una variable discreta u^* , que toma valores en un conjunto finito $\{r_1, \dots, r_L\}$ de números. Esta co-

rrrespondencia es, generalmente, una función escalera y la regla de cuantificación es como sigue: Definir $\{t_k, k = 1, \dots, L + 1\}$ como un conjunto de niveles de decisión o transición creciente con t_1 y t_{L+1} como los valores mínimo y máximo de u respectivamente. Si u cae en el intervalo $[t_k, t_{k+1})$, entonces se hace corresponder con r_k , el k -ésimo nivel de reconstrucción.

Los cuantificadores que se exponen en este trabajo cuantificadores *sin memoria*, que operan con un ejemplo cada vez y la salida depende sólo de la entrada. Estos cuantificadores son útiles en técnicas de codificación de imágenes como PCM, DPCM o codificación por transformación que se detallarán más adelante. Es importante ver que la correspondencia del cuantificador es irreversible. Es decir, que dada una determinada salida del cuantificador la entrada no puede ser determinada únicamente. Así, un cuantificador produce una distorsión que cualquier método razonable debe tratar de minimizar.

12.2.1. Cuantificador del óptimo cuadrado medio o de Lloyd-Max

Este cuantificador minimiza el error cuadrático medio para un número de niveles de cuantificación. Sea u una variable aleatoria escalar real con una distribución de probabilidad continua $p_u(u)$. Se desea encontrar los niveles de decisión t_k y los niveles de reconstrucción r_k para un cuantificador de L niveles de forma que su error cuadrático medio sea mínimo.

Usando el hecho de que $t_{k-1} \leq t_k$, las ecuaciones que resuelven el problema son:

$$t_k = \frac{(r_k + r_{k-1})}{2} \quad (12.3)$$

$$r_k = \frac{\int_{t_k}^{t_{k+1}} u p_u(u) du}{\int_{t_k}^{t_{k+1}} p_u(u) du} \quad (12.4)$$

Estas dos ecuaciones deben resolverse simultáneamente para los valores de t_k y t_{k+1} dados, lo que, en la práctica, se hace usando un método iterativo como el de Newton.

Las dos funciones de densidad más comúnmente usadas para cuantificación de imágenes son la gaussiana y la laplaciana.

12.2.2. El cuantificador uniforme optimal

Para distribuciones uniformes, las ecuaciones del cuantificador de Lloyd-Max se hacen lineales, dando niveles iguales entre los intervalos de reconstrucción y de transición. Este

cuantificador se llama también *cuantificador lineal* y sus ecuaciones son:

$$q = \frac{t_{L+1} - t_1}{L}, \quad t_k = t_{k-1} + q, \quad r_k = t_k + \frac{q}{2} \quad (12.5)$$

12.3. Codificación de pixels

En estas técnicas se procesa cada pixel de forma independiente, ignorando las dependencias entre pixels.

12.3.1. Modulación de códigos de pulso o PCM

En PCM la señal de entrada se muestrea, se cuantifica y codifica en un código apropiado. La salida del cuantificador se codifica, generalmente, mediante un código binario de longitud fija con B bits. Normalmente, 8 bits son suficientes para imágenes monocromas o de videoconferencia mientras que las imágenes médicas o en color necesitan entre 8 y 24 bits por pixel. Este proceso es el que habitualmente se realiza al captar una imagen real mediante un dispositivo digital.

12.3.2. Codificación por entropía

Si los pixels cuantificados no se distribuyen uniformemente, su entropía será menor que B y existirá un código que use menos de B bits por pixel. En la codificación por entropía la meta es codificar un bloque de M pixels, que tiene MB bits, con probabilidades p_i , $i = 0, \dots, L - 1$, $L = 2^{MB}$, con $-\log_2 p_i$ bits de forma que el número medio de bits sea H .

Para un tamaño de bloque dado, una técnica de codificación de longitud variable eficiente es la **codificación de Huffman**.

Algoritmo de codificación de Huffman

1. Ordenar las probabilidades de los símbolos, p_i , decrecientemente y considerarlas como nodos hojas de un árbol.
2. Mientras haya más de un nodo
 - Unir los dos nodos con menor probabilidad en un nuevo nodo con probabilidad la suma de las probabilidades de cada nodo.

- Asignar, arbitrariamente, un uno (1) a una de las ramas que llegan al nodo y un cero (0) a la otra.
3. Leer la secuencia de unos y ceros desde el nodo raíz al nodo hoja donde se encuentra el símbolo.

Este algoritmo da un *libro de códigos Huffman* para un conjunto de probabilidades dado. El tamaño del libro de códigos es L y el código más largo tiene como mucho L bits. El tamaño del libro de códigos es prohibitivamente grande cuando L crece. Una versión práctica es el *código de Huffman truncado* en el que, para un $L_1 < L$ convenientemente elegido, los primeros L_1 símbolos se representan por el código Huffman y el resto con un código de prefijo seguido de un código de longitud fija.

Otra alternativa es el llamado *código de Huffman modificado* en el que un entero i se representa como

$$i = qL_1 + j, \quad 0 \leq q \leq \text{Int} \left[\frac{L-1}{L_1} \right], \quad 0 \leq j \leq L_1 - 1 \quad (12.6)$$

Los primeros L_1 símbolos se representan por el código Huffman y el resto se representan por un código de prefijo que representa al cociente q , seguido de un *código terminador* que es el código Huffman para el resto j , $0 \leq j \leq L_1 - 1$.

Otra forma de codificación que trata de aproximarse al límite de la entropía es la **codificación aritmética**. Esta aproximación es conceptualmente diferente de la codificación de Huffman y no requiere códigos de longitud entera.

En la codificación aritmética los símbolos se ordenan en la recta real en el intervalo de probabilidad de 0 a 1 en una secuencia que es conocida tanto por el codificador como por el decodificador. Se asigna a cada símbolo un subintervalo igual a su probabilidad. Ya que la suma de las probabilidades de los símbolos suma uno, los subintervalos rellenan de forma precisa el intervalo de 0 a 1. En la figura 12.1 se muestra una posible ordenación de los símbolos según su probabilidad.

El objetivo de la codificación aritmética es crear una secuencia de códigos que es una fracción binaria apuntando al intervalo del símbolo que está siendo codificado. Así, si el símbolo es "símbolo 1" la secuencia de códigos es una fracción binaria mayor o igual al binario 0.01 (0.25 decimal) y menor que el binario 1.0. Si el símbolo es "símbolo 2" la secuencia de códigos es mayor que 0.001 en binario y menor que el binario 0.01. Si el símbolo es "símbolo 3" la secuencia de códigos es mayor que 0.0001 en binario y menor que el binario 0.001. Finalmente, si el símbolo es "símbolo 4" la secuencia de códigos es

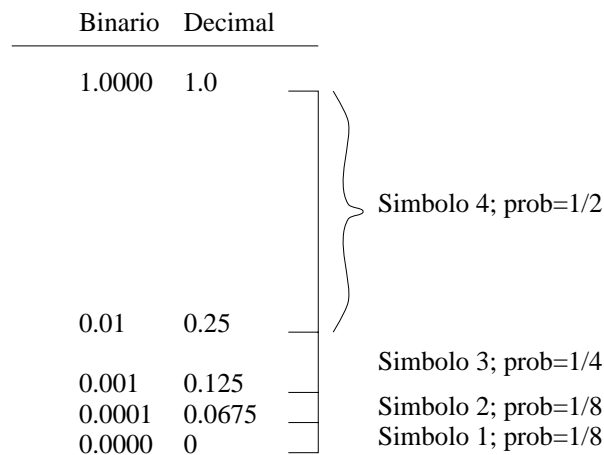


Figura 12.1: Particionamiento del intervalo $[0,1]$ de la recta real en subintervalos

mayor que 0 y menor que 0.0001 en binario. Si la secuencia de códigos sigue esta norma, el decodificador puede ver a que subintervalo está apuntando la secuencia de códigos y decodificar el símbolo apropiado.

Para la codificación de más símbolos es cuestión de ir subdividiendo el intervalo de probabilidad, siempre en proporción a la probabilidad de un determinado símbolo. Siempre y cuando sigamos las reglas y no dejemos que la secuencia de códigos apunten fuera del subintervalo asignado al símbolo, el decodificador podrá decodificar la secuencia.

Los límites entre intervalos se asignan siempre a uno de los intervalos. Aquí se ha escogido asignarlo al símbolo superior pero también es posible asignarlo al inferior.

Si tuvieramos que computar todos los posibles subintervalos para una secuencia larga de símbolos, la tarea computacional crecería exponencialmente. Afortunadamente la subdivisión se puede hacer recursivamente de forma que un intervalo se subdivide para cada símbolo codificado. Esta forma recursiva se conoce como codificación de Elias.

12.3.3. Codificación por longitud de recorridos (Run-Lenght Coding)

Considerar una fuente binaria cuya salida se codifica como el número de ceros entre dos unos, es decir, se codifica la longitud de los recorridos de ceros. Esto se llama *codificación por longitud de recorridos* (RLC). Si la probabilidad del 0 es p y tomamos m bits para representar el recorrido con un código de longitud fija, la máxima compresión que se puede

conseguir es

$$C = \frac{1 - p^M}{m(1 - p)} \quad (12.7)$$

donde $M = 2^m - 1$ y representa la longitud máxima de un recorrido.

12.3.4. Codificación de planos de bits

Si consideramos una imagen con 256 niveles de gris como un conjunto de ocho planos de 1 bit podemos codificar cada plano mediante RLC. Así se pueden conseguir ratios de compresión de 1.5 a 2. Este método es muy sensible a errores del canal.

12.4. Técnicas predictivas

La filosofía en que se basan las técnicas predictivas es eliminar la redundancia mutua entre pixels sucesivos y codificar sólo la información nueva. Considerar una secuencia muestreada $u(m)$ que ha sido codificada hasta $u(n - 1)$ y sean $u^*(n - 1)$, $u^*(n - 2)$, \dots , valores de la secuencia reproducida (decodificada). En $m = n$, cuando $u(n)$ llega, se predice una cantidad $\bar{u}^*(n)$, un estimador de $u(n)$, a partir de los ejemplos previamente decodificados $u^*(n - 1)$, $u^*(n - 2)$, \dots , es decir,

$$\bar{u}^*(n) = \Psi(u^*(n - 1), u^*(n - 2), \dots) \quad (12.8)$$

donde $\Psi(\dots)$ denota la regla de predicción. Esto es suficiente para codificar el *error de predicción*

$$e(n) = u(n) - \bar{u}^*(n) \quad (12.9)$$

Si $e^*(n)$ es el valor cuantificado de $e(n)$, entonces el valor reproducido de $u(n)$ sería

$$u^*(n) = \bar{u}^*(n) + e^*(n) \quad (12.10)$$

El proceso de codificación continua de esta forma. A este método se le llama *differential pulse code modulation* (DPCM) o PCM *diferencial*. Con un predictor razonable el valor cuadrático medio de la señal diferencia $e(n)$ es mucho más pequeño que el de $u(n)$. Esto significa que, para el mismo error cuadrático medio de cuantificación, $e(n)$ requiere menos bits de cuantificación que $u(n)$.

Un aspecto importante de DPCM se muestra en la ecuación (12.8), que dice que la predicción se basa en la salida – los ejemplos cuantificados – en lugar de basarse en la

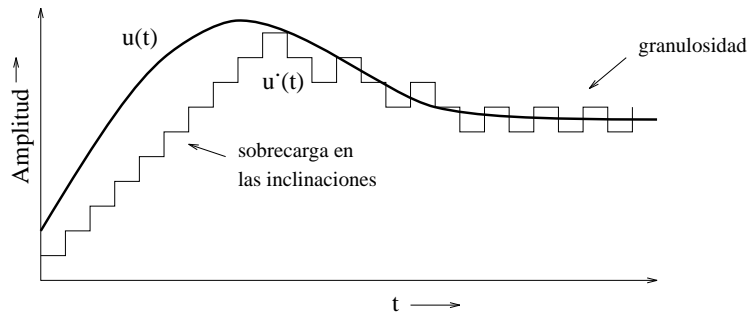


Figura 12.2: Salida típica para un codificador predictivo DM

entrada – los ejemplos no cuantificados. Esto hace que el error del cuantificador en un paso dado se realimente a la entrada del cuantificador en el siguiente paso lo que tiene un efecto estabilizador que previene la acumulación de error en la señal reconstruida $u^*(n)$.

12.4.1. Modulación Delta

Modulación Delta (*Delta Modulation* o *DM*) es el más simple de los codificadores predictivos. Usa una función retraso de un paso como predictor y un cuantificador de 1 bit, dando una representación de 1 bit de la señal. Así

$$\bar{u}^*(n) = u^*(n), \quad e(n) = u(n) - u^*(n-1) \quad (12.11)$$

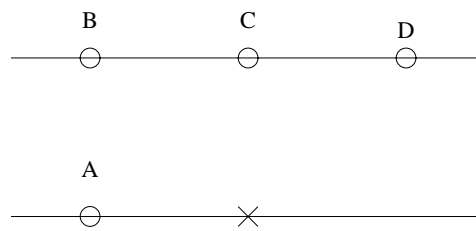
Las principales limitaciones de DM, como se observa en la figura 12.2, son (1) sobrecarga en las inclinaciones, (2) ruido de granulosidad e (3) inestabilidad por los errores canal.

La sobrecarga en las inclinaciones se producen cuando hay un salto grande o una discontinuidad en la señal a la que el cuantificador sólo puede responder en varios pasos delta. El ruido de granulosidad es la naturaleza de saltos de la salida cuando la señal de entrada es casi constante.

12.4.2. DPCM línea a línea

En este método cada línea de la imagen se codifica independientemente mediante la técnica DPCM. Generalmente se usa una representación autoregresiva (AR) adecuada para diseñar el predictor. El sistema de ecuaciones DPCM son:

$$\text{Predictor: } \bar{u}^*(n) = \sum_{k=1}^p a(k)u^*(n-k) \quad (12.12)$$

Figura 12.3: Pixels A , B , C y D usados en predicción bidimensional

$$\text{Entrada al cuantificador: } e(n) = u(n) - \bar{u}^*(n) \quad (12.13)$$

$$\text{Salida del cuantificador: } e^*(n) \quad (12.14)$$

$$\text{Filtro de reconstrucción: } u^*(n) = \bar{u}^*(n) + e^*(n) = \sum_{k=1}^p a(k)u^*(n-k) + e^*(n) \quad (12.15)$$

(salida reproducida)

12.4.3. DPCM bidimensional

La idea es extender el modelo anterior para la predicción en dos dimensiones. Así, para una imagen captada fila a fila el sistema de ecuaciones para DPCM bidimensional será:

$$\begin{aligned} \text{Predictor: } \bar{u}^*(m, n) = & a_1(k)u^*(m-1, n) + a_2(k)u^*(m, n-1) \\ & + a_3(k)u^*(m-1, n-1) + a_4(k)u^*(m-1, n-1) \end{aligned} \quad (12.16)$$

$$\text{Entrada al cuantificador: } e(m, n) = u(m, n) - \bar{u}^*(m, n) \quad (12.17)$$

$$\text{Salida del cuantificador: } e^*(m, n) \quad (12.18)$$

$$\text{Filtro de reconstrucción: } u^*(m, n) = \bar{u}^*(m, n) + e^*(m, n) \quad (12.19)$$

Varios parámetros posibles para el predictor se muestran en la tabla 12.1. Las letras A , B , C y D hacen referencia a los pixels que se muestran en la figura 12.3.

12.4.4. Técnicas Adaptativas

La idea general es mejorar el rendimiento de DPCM adaptando las características del predictor y del cuantificador a variaciones en las estadísticas locales de la imagen. Se diseñan principalmente para mejorar la calidad subjetiva de la imagen, especialmente en las aristas. Se basan en las dos aproximaciones siguientes.

Predictor	Comentario
media líneal al cuadrado	Determinada a partir de correlaciones en la imagen. Va muy bien ya que la clase imagen no cambia mucho.
Elemento anterior (γA)	Emborrona las aristas verticales o diagonales.
Predicción por media $\gamma \left(\frac{A+D}{2} \right)$ $\gamma \left(\frac{A+C}{2} \right)$ $\gamma \left(\frac{A+(C+D)/2}{2} \right)$	Mejora significativa sobre la predicción de elemento anterior para aristas verticales y con gran inclinación. Emborrona las aristas horizontales y con subida gradual. Los dos predictores que usan D son igualmente buenos pero mejores que $(A + C)/2$ en las aristas con subida gradual.
Predicción planar $\gamma(A + (C - B))$ $\gamma \left(A + \frac{(D-B)}{2} \right)$	Mejor que la predicción de elemento anterior pero peor que la predicción por media en lo que respecta a a errores del canal.
<i>leak</i> (γ)	$0 < \gamma < 1$. Conforme el <i>leak</i> se incrementa los errores de transmisión se hacen menos visibles pero la granularidad y el <i>contouring</i> (pixels con el mismo nivel de color debido a la cuantificación) se hacen mas visible.

Cuadro 12.1: Posibles parámetros para el predictor.

Control de ganancia adaptativo

Para un predictor fijo, la varianza del error de predicción variará con cambios en los detalles espaciales de la imagen. Un cuantificador adaptativo simple actualiza la varianza del error de predicción en cada paso y ajusta el espaciado de los niveles del cuantificador consecuentemente. Esto se puede hacer normalizando el error de predicción por su desviación típica actualizada y diseñado los niveles del cuantificador para entradas de varianza 1.

Clasificación adaptativa

Los esquemas de clasificación adaptativa segmentan la imagen según su actividad, o detalle espacial, y usa un cuantificador con características diferentes para cada clase de actividad. La varianza de los pixels en un vecindario del pixel a ser predicho es una medida simple de la actividad. Así, las regiones planas se cuantifican más finamente que las aristas o áreas con muchos detalles. Este esquema saca ventaja del hecho de que la visibilidad del ruido decrece cuando aumenta la actividad. Normalmente sólo son necesarias hasta cuatro clases de actividad.

12.5. Teoría de codificación por transformación

12.5.1. Codificador por transformación óptimo

La codificación por transformación, también llamada *cuantificación de bloques*, es una alternativa a la codificación predictiva. Un bloque de datos se transforma unitariamente de forma que una gran parte de su energía total se empaqueta en unos pocos coeficientes de la transformación, que son cuantificados independientemente. El codificador por transformación óptimo se define como aquel que minimiza la distorsión cuadrática media de la imagen reproducida para un número total de bits dado. Y resulta que la transformada KL lo es.

Suponiendo un vector, u , $N \times 1$ aleatorio con media 0 y covarianza R se transforma linealmente por una matriz A (compleja) $N \times N$, no necesariamente unitaria, para producir un vector (complejo), v , de forma que sus componentes $v(k)$ sean mutuamente decorrelados. Tras cuantificar cada coeficiente $v(k)$ independientemente, el vector de salida, v^* , se transforma linealmente por una matriz B para obtener un vector u^* . El pro-

blema es encontrar las matrices A y B óptimas y el cuantificador óptimo para minimizar la distorsión media, D , definida como:

$$D = \frac{1}{N} E \left[\sum_{n=1}^N (u(n) - u^*(n))^2 \right] = \frac{1}{N} E [(u - u^*)^T (u - u^*)]. \quad (12.20)$$

La solución a este problema se resume como sigue.

1. Para un cuantificador arbitrario la matriz óptima de reconstrucción, B , viene dada por

$$B = A^{-1} \Gamma \quad (12.21)$$

donde Γ es una matriz diagonal de elementos γ_k definida como

$$\gamma_k = \frac{\tilde{\lambda}_k}{\lambda_k^*} \quad (12.22)$$

$$\tilde{\lambda}_k = E [v(k) v^*(k)], \quad \lambda_k^* = E [|v^*(k)|^2] \quad (12.23)$$

2. El cuantificador de Lloyd-Max para cada $v(k)$ minimiza el error cuadrático medio global dando

$$\Gamma = I \quad (12.24)$$

3. La matriz de decorrelación óptima A es la transformación KL de u , es decir, las filas de A son los autovectores ortonormalizados de la matriz de autocovarianza R . Esto nos da

$$B = A^{-1} = A^{*T} \quad (12.25)$$

Notas

Ya que la transformación KL no es una transformación rápida, se reemplaza bien por una transformación unitaria como la coseno, seno, Fourier, Hadamard o Slant, que no son perfectas decorreladoras, o bien por una transformación decorreladora que no sea unitaria. En la práctica la primera opción da mejor rendimiento.

12.5.2. Codificación de imágenes por transformación

La teoría de codificación por transformación unidimensional puede ser extendida fácilmente a dos dimensiones convirtiendo la imagen $N \times M$ dada, $u(m, n)$, al vector $NM \times 1$ unidimensional u . La transformada KL de u sería una matriz $MN \times MN$. Como se ha

dicho antes, en la práctica esta transformación se reemplaza por la seno, coseno, Fourier, Slant o Hadamadar que empaquetan una considerable cantidad de la imagen en un pequeño número de coeficientes.

Para hacer la codificación por transformación práctica, la imagen dada se divide en pequeños bloques rectangulares y cada uno de esos bloques se codifica independientemente. Esto hace que el número de operaciones que se necesita para calcular la transformada se reduzca pero, conforme el tamaño del bloque se hace menor, el ratio de compresión también decrece.

12.6. Codificación híbrida y DPCM de vector

12.6.1. Idea básica

Si se capta la imagen por vectores, por ejemplo, una columna cada vez, es posible generalizar las técnicas DPCM considerando predictores recursivos de vectores. La codificación híbrida es un método para implementar un codificador DPCM de vectores $N \times 1$ mediante N codificadores DPCM escalares independientes. Esto se consigue combinando técnicas de codificación por transformación y predictivas. Normalmente la imagen se transforma unitariamente en una de sus dimensiones para decorrelar los pixels en esa dirección. Cada coeficiente de la transformación se codifica secuencialmente en la otra dirección mediante DPCM unidimensional. Esta técnica combina las ventajas de la simplicidad hardware de DPCM y el rendimiento robusto de la codificación por transformación.

12.6.2. Algoritmo de codificación híbrida

Sea u_n , $n = 0, 1, \dots$, un conjunto de $N \times 1$ columnas de una imagen que son transformadas como $v_n = \Psi u_n$, $n = 0, 1, \dots$. Para cada k , la secuencia $v_n(k)$ se modeliza, usualmente mediante un proceso AR de primer orden como

$$v_n = a(k)v_{n-1}(k) + b(k)e_n(k), \quad 1 \leq k \leq N, \quad n \geq 0 \quad (12.26)$$

Los parámetros de este modelo pueden ser identificados a partir de las covarianzas de $v_n(k)$, $n = 0, 1, \dots$, para cada k . Las ecuaciones DPCM para el k -ésimo canal puede ser escrita como

$$\text{Predictor: } \bar{u}^*(n) = a(k)v_{n-1}^*(k) \quad (12.27)$$

$$\text{Entrada al cuantificador: } \tilde{e}_n(k) = \frac{v_n(k) - \bar{v}_n^*(k)}{b(k)}, \quad (12.28)$$

$$\text{Salida del cuantificador: } \tilde{e}_n^*(k) \quad (12.29)$$

$$\text{Filtro de reconstrucción: } u_n^*(k) = \bar{v}_n^*(k) + b(k) \tilde{e}_n^*(k) \quad (12.30)$$

El receptor más simple reconstruye los vectores transformado según este filtro de reconstrucción y realiza la transformación inversa Ψ^{-1} . Teóricamente, la transformación Ψ debería ser la transformación KL de u_n . En la práctica se usa una transformación sinusoidal rápida como la seno o la coseno.

12.6.3. Codificación híbrida adaptativa

Actualizando los parámetros del modelo AR con variaciones en las estadísticas de la imagen se pueden diseñar algoritmos de codificación híbrida adaptativa. Uno de los métodos más sencillos es aplicar el algoritmo de control de ganancia adaptativo aunque el método de clasificación adaptativa da mejores resultados, especialmente en ratios bajos (1 bit/píxel).

Generalmente la codificación híbrida adaptativa puede mejorar las técnicas no adaptativas en unos 3dB, lo que es significativo a ratios bajos.

12.7. Codificación de imágenes en color y multiespectrales

Las técnicas de compresión discutidas anteriormente se pueden generalizar para imágenes en color como se muestra en la figura 12.4. Cada píxel se representa mediante un vector $p \times 1$. En el caso de las imágenes en color, la entrada es un vector 3×1 que contiene, normalmente, los componentes R , G y B , rojo, verde y azul respectivamente. Este vector se transforma a otro sistema de coordenadas de color, YIQ o YUV normalmente, donde cada componente puede ser procesado por un codificador espacial independiente. La transformación de RGB a YUV es inmediata mediante la aplicación de las siguientes ecuaciones:

$$Y = 0,3R + 0,6G + 0,1B \quad (12.31)$$

$$U = B - Y \quad (12.32)$$

$$V = R - Y \quad (12.33)$$

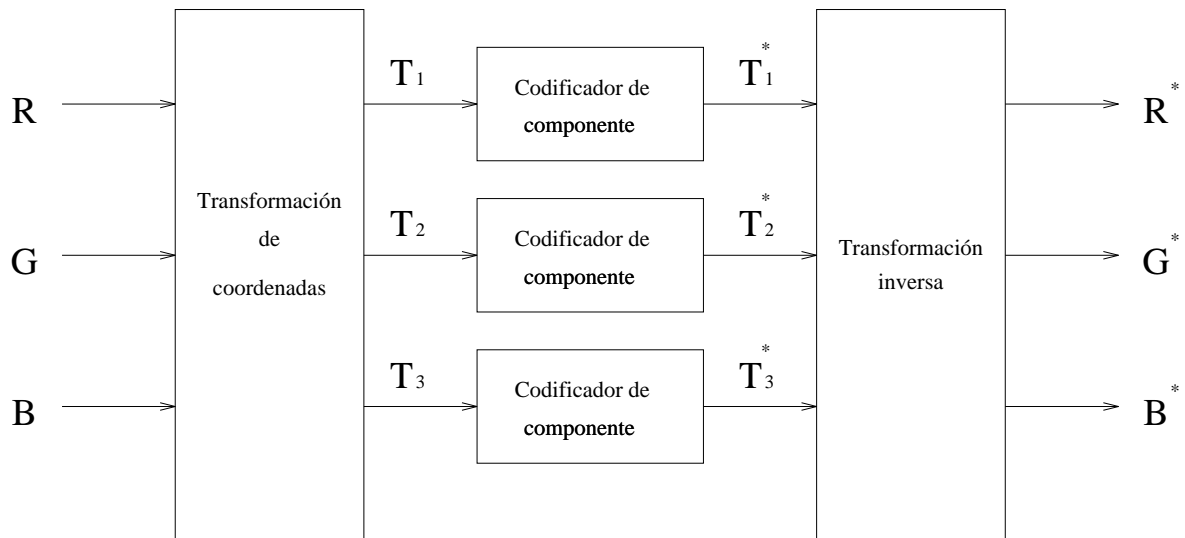


Figura 12.4: Codificación de componentes de imágenes en color.

Este espacio de color es el usado en el sistema de televisión europea. El sistema YIQ es el usado en el sistema de televisión norteamericana y se relaciona con el sistema YUV como sigue: Y es la misma para los dos espacios de color e I y Q se relacionan con U y V según las siguientes ecuaciones:

$$I = 0,74V - 0,27U \quad (12.34)$$

$$Q = 0,48V + 0,41U \quad (12.35)$$

Para la codificación de imágenes en color se debe tener en consideración que el componente de luminancia, Y , que representa la información acromática o imagen en escala de grises, suele tener un ancho de banda mayor que los componentes de crominancia, I y Q o U y V , que representa la información de color de la imagen. Esto quiere decir que los componentes de crominancia suelen tener menor número de pixels que el componente de luminancia. El segundo punto a tener en cuenta es que la métrica de las diferencias de color no es euclídea, es decir, que la misma potencia de ruido en diferentes componentes de color se percibe de forma diferente.

Un método de codificación alternativo para imágenes en color es procesar la señal de color compuesta. Esto es útil en aplicaciones de difusión en las que es desable manejar una sola señal.

La tabla 12.2 muestra los ratios de bits por pixels para diferentes algoritmos para las imágenes reproducidas con calidad de difusión. Estos resultados muestran que los compo-

Método	Componentes codificados	Descripción	Ratio por componente bits/componente/pixel	Ratio medio bits/pixel
PCM	R, G, B	Datos sin transformar	8	24
	Y, I, Q	I, Q submuestreados	8	12
DPCM con retraso de un paso (DM)	Y, I, Q	I, Q submuestreados	2 a 3	2 a 4.5
Transformación (coseno, slant)	Y, I, Q	no submuestreado	Y (1.75 a 2)	2.5 a 3
	Y, U, V		I, Q (0.75 a 1)	
	Y, I, Q	Igual que arriba con clasificación adaptativa	Variable	1 a 2

Cuadro 12.2: Ratios en bits por pixel para diferentes algoritmos de codificación de imágenes en color con calidad de difusión

mentos de crominancia pueden ser codificados desde $\frac{1}{2}$ bit por pixel (mediante codificación transformacional adaptativa) a 1 bit por pixel (mediante DPCM).

Dada la flexibilidad en el diseño de codificadores, la codificación de componentes da mejor rendimiento que la codificación de la señal compuesta.

12.8. El estándar de compresión de imágenes estáticas JPEG

12.8.1. Introducción

Hemos visto que mediante las técnicas descritas podemos comprimir la imagen entre $1/10$ y $1/50$ de su tamaño descomprimido sin afectar visiblemente la calidad de la imagen. Pero las técnicas de compresión por sí solas no pueden ganarse el creciente mercado que se abre a las imágenes. Por ello se hace necesario un estándar para que diferentes sistemas puedan compartir esas imágenes,

Durante los últimos años se ha hecho un esfuerzo de estandarización conocido con el acrónimo de JPEG (**J**oint **P**hotographic **E**xperts **G**roup) (Grupo Conjunto de Expertos Fotográficos) que ha dado como resultado el primer estándar internacional para compresión de imágenes digitales multinivel, tanto en escala de grises como en color. El conjunto”en

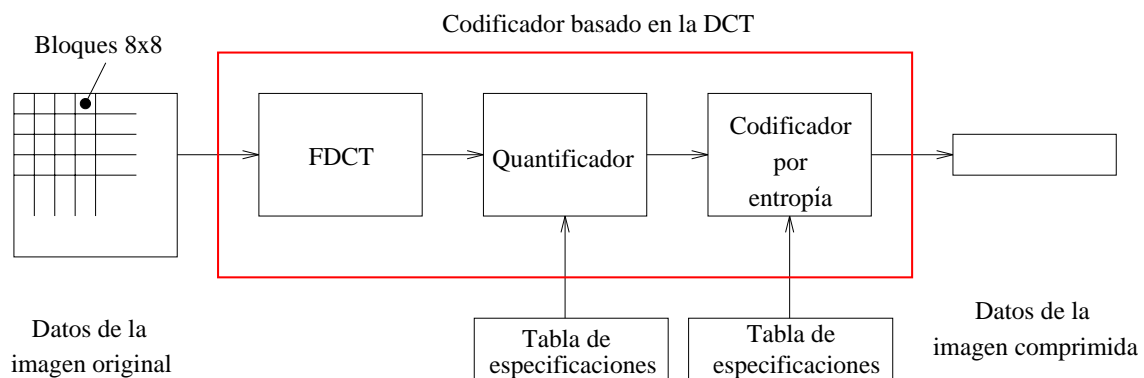


Figura 12.5: Pasos del proceso de codificación basado en la DCT

el nombre viene de la cooperación entre CCITT e ISO. De esta forma JPEG es tanto un estándar ISO como una recomendación CCITT.

JPEG escogió entre todos los métodos de compresión propuestos un método basado en codificación por transformación usando la transformada coseno discreta (DCT) 8×8 pues produjo la mejor relación calidad/compresión.

12.8.2. Pasos del proceso para codificación basado en la DCT

Las figuras 12.5 y 12.6 muestran los principales pasos de una codificación basada en la DCT. Las figuras ilustran el caso especial de la compresión una imagen en escala de grises. La compresión de una imagen en escala de grises puede verse como la compresión de una secuencia de bloques 8×8 que forman la imagen. Se puede pensar que la codificación de una imagen en color puede ser, aproximadamente, la codificación de múltiples imágenes en escala de grises que son comprimidas una cada vez o comprimida alternativamente entrelazando los bloques 8×8 de cada banda.

8×8 FDCT y IDCT

A la entrada del codificador, la imagen original se agrupa en bloques 8×8 , se translada de enteros sin signo en el rango $[0, 2^P - 1]$ a enteros con signo con rango $[-2^{P-1}, 2^{P-1} - 1]$ y se le da como entrada a la transformada DCT directa (*Forward DCT* o FDCT). En la salida del decodificador, la transformada DCT inversa (*Inverse DCT* o IDCT) devuelve bloques 8×8 para formar la imagen reconstruida. Las siguientes ecuaciones son

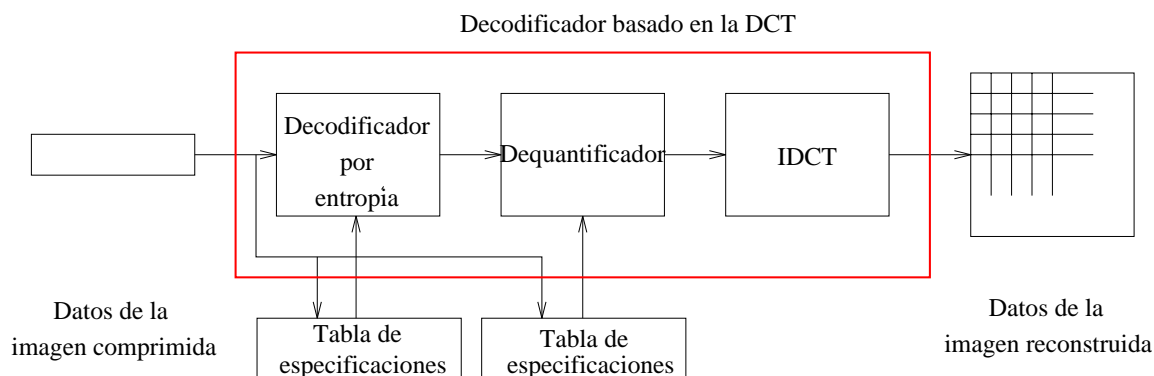


Figura 12.6: Pasos del proceso de decodificación basado en la DCT

las definiciones matemáticas idealizadas de la FDCT 8×8 y de la IDCT 8×8 :

$$F(u, v) = \frac{C(u)}{2} \frac{C(v)}{2} \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (12.36)$$

$$f(x, y) = \sum_{u=0}^7 \frac{C(u)}{2} \sum_{v=0}^7 \frac{C(v)}{2} F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \quad (12.37)$$

$$\text{donde: } C(u), C(v) = \begin{cases} 1/\sqrt{2} & \text{para } u, v=0 \\ 1 & \text{en otro caso} \end{cases} \quad (12.38)$$

La FDCT toma cada bloque 8×8 y lo descompone en 64 señales de una base ortogonal. Cada una de estas señales contiene una de las 64 "frecuencias espaciales" 2D únicas que comprenden el "espectro" de la señal de entrada. La salida de la FDCT es un conjunto de 64 amplitudes de señal de la base o coeficientes de la DCT cuyos valores son únicamente determinados por una señal de 64 puntos de entrada particular. Al coeficiente con frecuencia cero en las dos dimensiones se le llama coeficiente DC a los otros 63 coeficientes, coeficientes AC". Estos nombres vienen del uso habitual de la DCT en electrónica para analizar circuitos donde hay simultáneamente términos de corriente continua (**D**irect **C**urrent) y de corriente alterna (**A**lter **C**urrent).

Dado que los valores de la imagen varían lentamente entre píxeles consecutivos, el proceso de la FDCT realiza la compresión de los datos concentrando la mayor parte de la señal en las frecuencias espaciales bajas. Para un bloque típico 8×8 de una imagen típica, la mayoría de las frecuencias espaciales tienen una amplitud cero o casi cero y no serán codificadas.

En el decodificador la IDCT realiza el proceso inverso. Toma 64 coeficientes DCT (que para entonces ya han sido cuantificados) y reconstruye una imagen de 64 puntos de salida sumando las señales base. Si la FDCT y la IDCT pudieran ser calculadas con precisión absoluta y los coeficientes de la DCT no se cuantificaran se podría reconstruir perfectamente la imagen original. En principio, la DCT no introduce pérdidas en la imagen original, simplemente la transforma a un dominio en el que puede ser codificada más eficientemente.

Cuantificación

Después de salir de la FDCT, cada uno de los 64 coeficientes DCT se cuantifica uniformemente en conjunción con una tabla de cuantificación de 64 elementos, que debe ser especificada por la aplicación o el usuario como entrada al codificador. Cada elemento de la tabla de cuantificación puede ser cualquier valor entero entre 1 y 255, que especifica el tamaño del paso del cuantificador para su coeficiente DCT correspondiente. El propósito de la cuantificación es conseguir aún más compresión representando los coeficientes DCT sin más precisión que la que es necesaria para conseguir la calidad de imagen deseada. Visto de otra forma, lo que se pretende con este proceso es descartar la información que no es visualmente relevante. La cuantificación es una aplicación muchos a uno y, por tanto, es fundamentalmente con pérdidas. De hecho es la principal fuente de pérdidas en los codificadores basados en la DCT.

La cuantificación se define como una división de cada coeficiente DCT por su coeficiente de cuantificación, es decir, el tamaño del paso de la cuantificación, correspondiente, redondeado al entero más cercano:

$$F^*(u, v) = \text{Redondeo} \left(\frac{F(u, v)}{Q(u, v)} \right) \quad (12.39)$$

El valor de salida está normalizado por el tamaño del paso de cuantificación. La de-cuantificación es la función inversa que, en este caso, significa simplemente que la normalización se elimina multiplicando por el tamaño del paso, lo que devuelve el resultado a la representación apropiada para la entrada de la IDCT:

$$F'(u, v) = F^*(u, v) * Q(u, v) \quad (12.40)$$

Cuando se quiere comprimir una imagen tanto como sea posible sin artificios visibles hace falta escoger cada tamaño del paso de cuantificación como el umbral perceptual para

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Cuadro 12.3: Tabla de cuantificación de la luminancia propuesta por ISO

la contribución de su función base coseno correspondiente. Estos umbrales son también funciones de las características de la imagen original, de las características del visualizador y de la distancia de visión.

La tabla 12.3 fue probada por los miembros de JPEG y apareció en el standard ISO como informativa, no como un requerimiento, para cuantificar la luminancia de una imagen.

Codificación DC y secuencia Zig-Zag

Tras la cuantificación, el coeficiente DC se trata de forma separada de los 63 coeficientes AC. El coeficiente DC es una medida del valor medio de los 64 pixels de la imagen. Ya que normalmente hay un fuerte correlación entre los coeficientes DC de dos bloques 8×8 adyacentes, los coeficientes DC se codifican como la diferencia entre el término DC del bloque previo en el orden de codificación, como se muestra en la figura 12.7. Este tratamiento especial merece la pena puesto que los coeficientes DC contienen una fracción significativa de la energía total de la imagen.

Finalmente, todos los coeficientes cuantificados se ordenan en una secuencia en zig-zag, como se muestra en la figura 12.8. Esta ordenación ayuda a facilitar la codificación por entropía colocando los coeficientes de las bajas frecuencias, que normalmente no son cero, antes de los coeficientes de las altas frecuencias.

Codificación por entropía

El paso final de la codificación basada en la DCT es la codificación por entropía. Este paso consigue una compresión sin pérdidas adicional al codificar los coeficientes cuantifica-

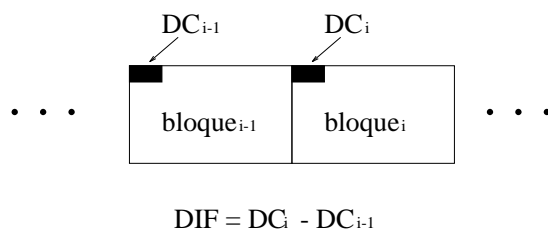


Figura 12.7: Codificación diferencial de los coeficientes DC

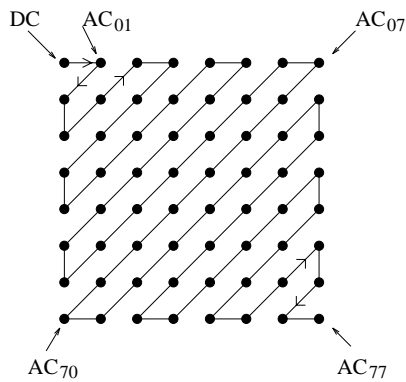


Figura 12.8: Secuencia zig-zag para la preparación de los coeficientes cuantificados para la codificación por entropía

dos de la DCT de una forma más compacta basándose en sus características estadísticas. La propuesta de JPEG especificaba dos métodos de codificación por entropía: Codificación de Huffman y codificación aritmética. El modelo base de codificador–decodificador (codec) usa códigos Huffman pero se especifican codecs con los dos métodos para todos los modos de operación.

Es útil considerar la codificación por entropía como un proceso de dos pasos. El primero convierte la secuencia de coeficientes cuantificados en zig-zag en una secuencia de símbolos intermedia mediante RLC. El segundo paso convierte los símbolos a una secuencia de datos en la que los símbolos no tienen ya fronteras externamente identificables. La forma y definición de los símbolos intermedios depende tanto del modo de operación como del método de codificación por entropía.

La codificación de Huffman requiere que la aplicación especifique uno o más conjuntos de tablas de códigos Huffman. Éstas deben ser las mismas para comprimir una imagen y para descomprimirla. Las tablas de códigos Huffman pueden ser predefinidas por la aplicación o bien calculadas específicamente para una imagen.

El método de codificación aritmética propuesto por JPEG, por el contrario, no requiere ninguna tabla externa porque es capaz de adaptarse a las características de la imagen conforme la va codificando. Este tipo de codificación produce entre un 5 % y un 10 % más de compresión que los códigos Huffman para muchas de las imágenes de prueba probadas por los miembros de JPEG pero es más complejo de implementar que la codificación Huffman.

Codificación de imágenes en color

El modelo de imagen usado en la propuesta de JPEG es una abstracción de diferentes tipos de imágenes y aplicaciones y sólo contiene lo que es necesario para comprimir y reconstruir una imagen digital. Así debe reconocerse que el formato comprimido de JPEG no codifica suficiente información para ser una representación completa de la imagen. Por ejemplo, JPEG no especifica o codifica información sobre la razón de aspecto de los pixels, el espacio de color o las características de adquisición de la imagen.

Una imagen fuente puede contener desde 1 a 255 componentes, a veces llamadas color o bandas espectrales o canales. Cada componente es una matriz rectangular de elementos. Cada elemento se define como un entero sin signo con P bits de precisión con un valor en el rango $[0, 2^P - 1]$. Todos los elementos de todas las componentes en la misma imagen

deben tener la misma precisión P.

El i -ésimo componente tiene unas dimensiones x_i por y_i . Para acomodarse a formatos en los que algunas componentes están muestreadas a diferentes razones que otras, las componentes pueden tener dimensiones diferentes. Las dimensiones deben tener una relación mútua, definida por H_i y V_i , los factores de muestreo relativo horizontal y vertical, que deben ser especificados para cada componente. Las dimensiones totales, X e Y se definen como el máximo x_i e y_i de todas las componentes de la imagen y puede ser cualquier número hasta 2^{16} . H y V sólo pueden ser enteros entre 1 y 4. Los parámetros codificados son X , Y y H_i y V_i para cada componente. La decodificación reconstruye las dimensiones de x_i e y_i para cada componente de acuerdo a la relación mostrada en la ecuación 12.41:

$$\begin{aligned} x_i &= \lceil X \times \frac{H_i}{H_{max}} \rceil \\ y_i &= \lceil Y \times \frac{V_i}{V_{max}} \rceil \end{aligned} \quad (12.41)$$

donde $\lceil \rceil$ representa la función redondeo por exceso.

Para la codificación de las bandas, JPEG propuso la realización de un entrelazado generalizado de las bandas consistente, básicamente, en entrelazar bloques 8×8 codificados de cada banda en lugar de entrelazar pixels. Además JPEG permite que algunos componentes se entrelacen y otros no lo hagan en la misma imagen comprimida.

Además de controlar el entrelazado o no entrelazado los componentes, los codecs JPEG deben controlar la aplicación de las tablas de datos adecuadas al componente adecuado. Se tiene que usar la misma tabla de cuantificación y tabla o tablas de codificación por entropía a todos los elementos de cada componente. Ya que JPEG puede almacenar varias tablas de cuantificación y de codificación por entropía también debe tener mecanismos para cambiar de tabla durante la decodificación de una imagen que contenga varios componentes, posiblemente entrelazados, para así poder aplicar la tabla adecuada al componente adecuado.

Compresión y calidad de la imagen

Para imágenes en color con escenas moderadamente complejas todos los modos de operación basados en la DCT producen los siguientes niveles de calidad de imagen en función de los niveles de compresión. Estos niveles son sólo guías ya que la calidad y la compresión pueden variar mucho dependiendo de las características de la imagen. Las

unidades usadas son bits/píxel obtenidos como la media del número total de bits de la imagen comprimida dividido entre el número de píxeles de la imagen.

- 0.25–0.5 bits/píxel: calidad moderada a buena, suficiente para algunas aplicaciones;
- 0.5–0.75 bits/píxel: calidad buena a muy buena, suficiente para muchas aplicaciones;
- 0.75–1.5 bits/píxel: calidad excelente, suficiente para la mayoría de las aplicaciones;
- 1.5–2.0 bits/píxel: normalmente indistinguible de la original, suficiente para las aplicaciones que requieren máxima calidad.

12.8.3. Modelo Base y otros codecs secuenciales basados en la DCT

El modo de operación secuencial propuesto por JPEG consiste en la FDCT y los pasos de la cuantificación descritos en la sección 12.8.2. Además del codec base, se definen otros tipos de codecs para dos diferentes tipos de precisión (8 y 12 bits) y dos tipos diferentes de codificación por entropía (Huffman y aritmética).

La codificación secuencial base se usa para imágenes con 8 bits por píxel para cada banda y usa solamente codificación Huffman. Se diferencia también de otros codecs en que sólo puede almacenar dos tablas de códigos Huffman (normalmente una para los coeficientes AC y otra para los DC). Esto significa que si hay varias bandas, éstas deben compartir las tablas de códigos.

Para las aplicaciones que usan componentes de color esta limitación raramente lo es porque se usan espacios de colores como YUV que representan la información cromática (el color) en dos bandas y la información acromática (escala de grises) en una tercera y consiguen mejor compresión que la escala RGB. De esta forma se puede usar una tabla de códigos Huffman para la componentes acromática y otra para las componentes cromáticas. Ya que las estadísticas de los coeficientes DCT son similares para las componentes de crominancia de la mayoría de las imágenes una tabla de códigos Huffman puede codificarlas casi tan óptimamente como dos.

Apéndice A

Modelos de Visión

Cuando comienza la reflexión filosófica ésta se centra en la percepción del mundo exterior. El individuo se queda perplejo ante la existencia de un mundo de objetos que puede ver si mantiene los ojos abiertos y hay suficiente luz. Los objetos se mueven, también puede moverse el observador y sin embargo sigue viendo ahí un mundo estable y organizado. También percibimos formas y colores e incluso creemos que podemos percibir, cuando no hay objetos, dependiendo de nuestra memoria, nuestras imágenes mentales o incluso la fuerza de la imaginación.

Todo lo que hemos dicho parece obvio. Sin embargo, la historia de la Filosofía y Psicología nos demuestra lo difícil que es encontrar una explicación razonable de como se produce la percepción. ¿Toda la información necesaria para la percepción está en el mundo externo o aportamos conocimiento interno a la percepción?. Preguntas como ésta y otras muchas ya fueron estudiadas por los filósofos griegos. Los psicólogos también abordaron el problema, mientras que Hermann von Helmholtz pensaba que no había suficiente información en el estímulo exterior, J.J. Gibson argumentaba que los sentidos pueden extraer directamente del ambiente la información que es necesaria para sobrevivir.

Dos nuevas líneas de investigación entran en la discusión sobre la percepción. Por una parte la dedicada a determinar la sensibilidad de determinadas células nerviosas dando lugar a los modelos reduccionistas y la que a nosotros más nos ocupa, la que veremos en la sección siguiente

Con la llegada de los ordenadores, otro grupo de investigadores comenzó a analizar los procesos de percepción visual. Para estos investigadores el desafío no estaba en descubrir como se produce la percepción, la idea final era analizar como era posible la percepción *en*

cualquier clase de organismo o aparato mecánico. El objetivo era pues diseñar máquinas y programas que “vieran”.

Después de unos malos comienzos en los finales de los 50 y comienzo de los 60, estudiantes de percepción iniciaron una línea de trabajo que se denominó *análisis de escenas* ([4], [35]). Un programa de investigación típico tomaba una imagen, usualmente un dibujo, siendo el objetivo interpretar la imagen en función de los objetos que en ella había, establecer sus relaciones y describir los propios objetos. Así investigadores como David Waltz fueron capaces de crear programas que interpretaban dibujos que contenían sólo bloques. El problema era que tanto en este como en otros programas, los mecanismos usados no eran en absoluto psicológicamente plausibles.

Era obvio que si queríamos avanzar en el concepto de visión artificial era necesario considerar no solo la percepción de un tipo particular de imagen o micromundo, era necesario atacar el proceso de percepción a un nivel más fundamental, el objetivo debería ser producir un mecanismo que, *con independencia de su posible similitud con los usados por los seres animados*, pudiera al menos tratar todo el rango de escenas que se pueden percibir así como el abanico completo de tareas perceptuales que incluirían la percepción de movimiento, profundidad, textura y otras variables que hacen el sistema de visión humana tan potente. Era obvio que era el momento, era entonces posible combinar la psicología, las ciencias neurológicas y la inteligencia artificial. Era el momento para que apareciera el gran investigador de la visión, David Marr.

A.1. El trabajo de David Marr

Aunque la vida investigadora de David Marr fue muy corta, ésta fue extremadamente productiva. Formado en el campo de la neurofisiología en Cambridge, Inglaterra, comenzó su investigación con el estudio del funcionamiento del cerebellum. Pronto descubrió que las cuestiones más importantes sobre la conducta nunca podrían tener respuesta mirando sólo al cerebro. Como él decía, ni tan siquiera si conociéramos el funcionamiento de cada célula y cada conexión. Era como si intentásemos comprender el vuelo examinando sólo las alas de un pájaro. Para entender el problema de la percepción era necesaria una teoría que tuviera en cuenta los problemas que aparecen en la percepción de objetos, cómo las restricciones que aparecen pueden ser contraatacadas y cómo puede llevarse a cabo su implementación en mecanismos particulares que podrían ir de los ordenadores al

cerebro. Estas ideas estimularon a Marr para comenzar su trabajo en Inteligencia Artificial en el Instituto Tecnológico de Massachusetts desde 1973 hasta su muerte de leucemia en 1980.

En este breve periodo, Marr diseñó un programa sobre como aproximar la percepción visual en particular y de como estudiar sistemas de conocimiento en general. Un punto fundamental en su teoría fue la creencia de que *la visión es la construcción de descripciones simbólicas eficientes de los objetos que se encuentran en el mundo*.

A.1.1. Niveles del Análisis de Escenas

Marr sostenía que los procesos de la visión son sistemas complejos de procesamiento de información. Es obvio que el término proceso es muy amplio y que incluye como ejemplos la suma, la obtención de la transformada de Fourier o incluso ir de compras.

Es fundamental entender bien un proceso a diferentes niveles: el primer nivel, el más abstracto recibe el nombre de teoría de la computación y se refiere a qué se hace y por qué, el segundo nivel del proceso implica escoger dos cosas: una representación para la entrada y la salida del proceso y un algoritmo por medio del cual pueda entenderse la transformación, podría decirse que este segundo nivel del proceso se refiere al como del proceso. El tercer nivel se refiere al mecanismo en el cual ha de realizarse físicamente el proceso.

Veamos estas ideas con un ejemplo simple, la suma. La teoría de la computación nos diría que la suma es una proyección, que normalmente se indica por $+$, de pares de números en números individuales. Esta proyección tiene una serie de propiedades abstractas, que pueden y deben entenderse como restricciones, y que son conmutatividad, asociatividad, la existencia de elemento neutro y también la existencia de elemento simétrico. Pasemos ahora al segundo nivel. Como representación podemos escoger los números arábigos y como algoritmo el clásico de la suma. Aquí es importante señalar que hay una amplia posibilidad de representaciones y que muy seguramente el algoritmo a usar depende mucho de la representación. Llegamos ahora al tercer nivel, el mecanismo en el cual ha de realizarse físicamente el proceso, es obvio que la suma de dos números puede realizarse por un niño de derecha a izquierda, llevándose un dígito cuando sea necesario y éste puede ser el mismo algoritmo que emplee una caja registradora, pero es obvio que la realización física de ambos es bastante diferente. No hace falta hacer hincapie en la importancia de entender bien los conceptos de la teoría de la computación (ver [28] pg 36).

Volvamos a los tres niveles de descripción para los problemas de percepción visual. En este caso, los tres niveles están relacionados de un modo laxo y que algunos fenómenos sólo pueden ser explicados a determinados niveles. Por ejemplo, una observación psicofísica de como un estímulo afecta a la visión debe formularse al nivel apropiado. La psicofísica está relacionada más directamente con el nivel de algoritmo y representación. Diferentes algoritmos tienden a fracasar cuando se les lleva a los límites de su rendimiento o se les priva de su información crítica. La psicofísica también puede ayudarnos a determinar la naturaleza de la representación (ver [28] pg 35).

De igual modo es obvio que la neuroanatomía está vinculada principalmente al tercer nivel, el de la realización física del cálculo. Lo mismo ocurre con los mecanismos sinápticos, los potenciales de acción, las interacciones inhibitorias, etc. La neurofisiología, dedicada en una parte a la forma en que los procesos visuales se implementan en el cerebro también esta relacionada con este nivel.

La explicación de un proceso puede también realizarse a uno o varios niveles. La ambigüedad del cubo de Necker podría explicarse por una parte con la existencia de una red neural biestable pero también sería preciso mencionar la existencia de dos interpretaciones tridimensionales diferentes y totalmente plausibles de esta imagen bidimensional.

Veamos, una vez descritos los sistemas de procesamiento de la información en general, como concebía Marr los sistemas visuales más avanzados, para pasar a estudiar con más detalle su modelo en las secciones siguientes. Según cuenta el propio autor ([28] pg 42) su forma de enfocar estos problemas estuvo muy influida por una conferencia ofrecida por Elizabeth Warrington en el MIT en octubre de 1973, en esta conferencia se describían las capacidades y limitaciones de los pacientes que habían sufrido una lesión parietal izquierda o derecha. Aquellos pacientes que sufrían lesiones en el lado derecho podían reconocer un objeto común siempre que la perspectiva de la que gozaban de él fuera sencilla. No sólo reconocían el objeto, también sabían su nombre y sus aspectos semánticos. Si el punto de vista no era sencillo, por ejemplo un cubo de agua visto desde arriba, no lo reconocían e incluso negaban vehementemente que pudiera ser un cubo. Los enfermos con lesiones en el parietal izquierdo se comportaban de modo distinto. A menudo habían perdido el lenguaje, de modo que no podían nombrar el objeto observado o exponer su propósito y aspectos semánticos, pero percibían y expresaban su geometría incluso en perspectivas difíciles.

Era pues obvio, según el autor, que *en primer lugar la representación de la forma de un objeto está almacenada en un lugar diferente y, por tanto, es algo bastante distinto a*

la representación de su uso y propósito. En segundo lugar, por sí misma la visión puede arrojar una descripción interna de la forma de un objeto observado incluso cuando no se le reconoce en el sentido convencional de entender su uso y su propósito.

Llegamos así a la idea fundamental: la visión nos habla acerca de la forma, el espacio y la disposición espacial. El objetivo de la visión es pues la construcción de las formas y las posiciones de las cosas a partir de las imágenes.

Pero, aunque es muy deseable que la visión nos ofrezca una descripción completamente invariante de la forma a partir de una imagen, casi con toda seguridad será imposible conseguirlo en un único paso. El marco teórico que se describe en el trabajo de Marr divide la extracción de la información sobre la forma en tres etapas de representación que estudiaremos en las secciones siguientes: la representación de las propiedades de la imagen bidimensional, tales como los cambios de intensidad y la geometría local bidimensional (*el esbozo primitivo*); la representación de las propiedades de las superficies visibles en un sistema de coordenadas centrado en el observador, como la orientación de la superficie, la distancia que le separa del observador y las discontinuidades en estas cantidades, la reflectancia de la superficie y alguna descripción tosca de la iluminación reinante (*el esbozo $2\frac{1}{2} D$*), y una representación centrada en el objeto de la estructura tridimensional y de la organización de la forma observada, junto con alguna descripción de las propiedades de su superficie (*Representación de modelo 3 D*).

Es importante hacer notar que los detalles de la teoría de la visión de Marr no son necesariamente correctos, ver [36], sin embargo su aproximación es sin lugar a duda la de más éxito entre las aproximaciones recientes a la visión.

A.2. El Esbozo Primitivo

Los principales factores responsables de los valores de intensidad de una imagen son cuatro: (1) la geometría, (2) la reflectancia de las superficies visibles, (3) la iluminación de la escena y (4) el punto de observación. En una imagen todos estos factores se entremezclan (ver [50], cap. 1). El propósito del procesamiento visual inicial es el de descubrir cuales son los cambios que se deben a cada factor y crear representaciones en las que se distinga entre todos ellos.

Según Marr este objetivo se alcanza en dos etapas. Primero se obtienen representaciones adecuadas de los cambios y estructuras de la imagen, lo que implica operaciones

como la detección de cambios de intensidad, la representación y el análisis de la estructura geométrica local y la detección de ciertos efectos de iluminación como las fuentes de luz y la transparencia. Esta representación constituye como ya hemos dicho el *esbozo primario*. Sobre ella actúan varios procesos como la estereoscopia y el movimiento, (recordemos el concepto de proceso con tres pasos fundamentales), para derivar una representación todavía retinocéntrica de la geometría de las superficies visibles. Esta representación se denomina como ya sabemos *esbozo* $2\frac{1}{2}D$.

El algoritmo ideado por Marr y Hildreth [29] para la construcción del esbozo primario en bruto comienza con la transformación de la representación en niveles de gris en varias representaciones independientes, suavizadas a diferentes escalas, esto es así porque en las imágenes naturales los cambios de intensidad tienen lugar sobre una serie de escalas diferentes.

En resumen, Marr y Hildreth proponen que la imagen se pase a través de una serie de dos o más filtros gaussianos que sustituyan la matriz de intensidades en la imagen I por una serie de matrices de valores $G * I$ que denota la convolución de la imagen original con un filtro gaussiano. Los diferentes filtros gaussianos se obtienen con diferentes desviaciones típicas en la distribución de Gauss (ver figura A.1).

Con ello llegamos a la segunda operación que realiza el algoritmo de Marr y Hildreth: la localización de los cambios de intensidad en las múltiples representaciones de la imagen con diferente borrosidad. Esta se hace detectando los cruces por cero de la imagen que se obtiene aplicando el laplaciano ∇^2 a la imagen emborronada, es decir encontrando los ceros de $\nabla^2(G * I)$, observemos que para calcular donde hay variación más rápida en una función calculamos o bien los máximos de la primera derivada o los pasos por cero de la segunda (ver figura A.2).

Es obvio que es necesario combinar los pasos por cero a diferentes escalas. Marr y Hildreth [29] sugieren un paso final en el que se combinen la información sobre los pasos por cero de cada filtro $\nabla^2 G$ por separado.

La primera regla para la combinación de las salidas de los filtros $\nabla^2 G$ que construyan el esbozo primario en bruto es que se coloque en el esbozo un *segmento de borde* allí donde los segmentos de pasos de cero de filtros de similar anchura se emparejen. En algunas situaciones, un segmento de pasos de cero en un canal ancho puede emparejarse por otros dos paralelos de contraste opuesto en un canal estrecho; esta situación se representa en el esbozo primario en bruto por una *barra*. Los extremos de las barras se representan mediante *terminaciones*, y los bucles cerrados de los elementos de borde se representan

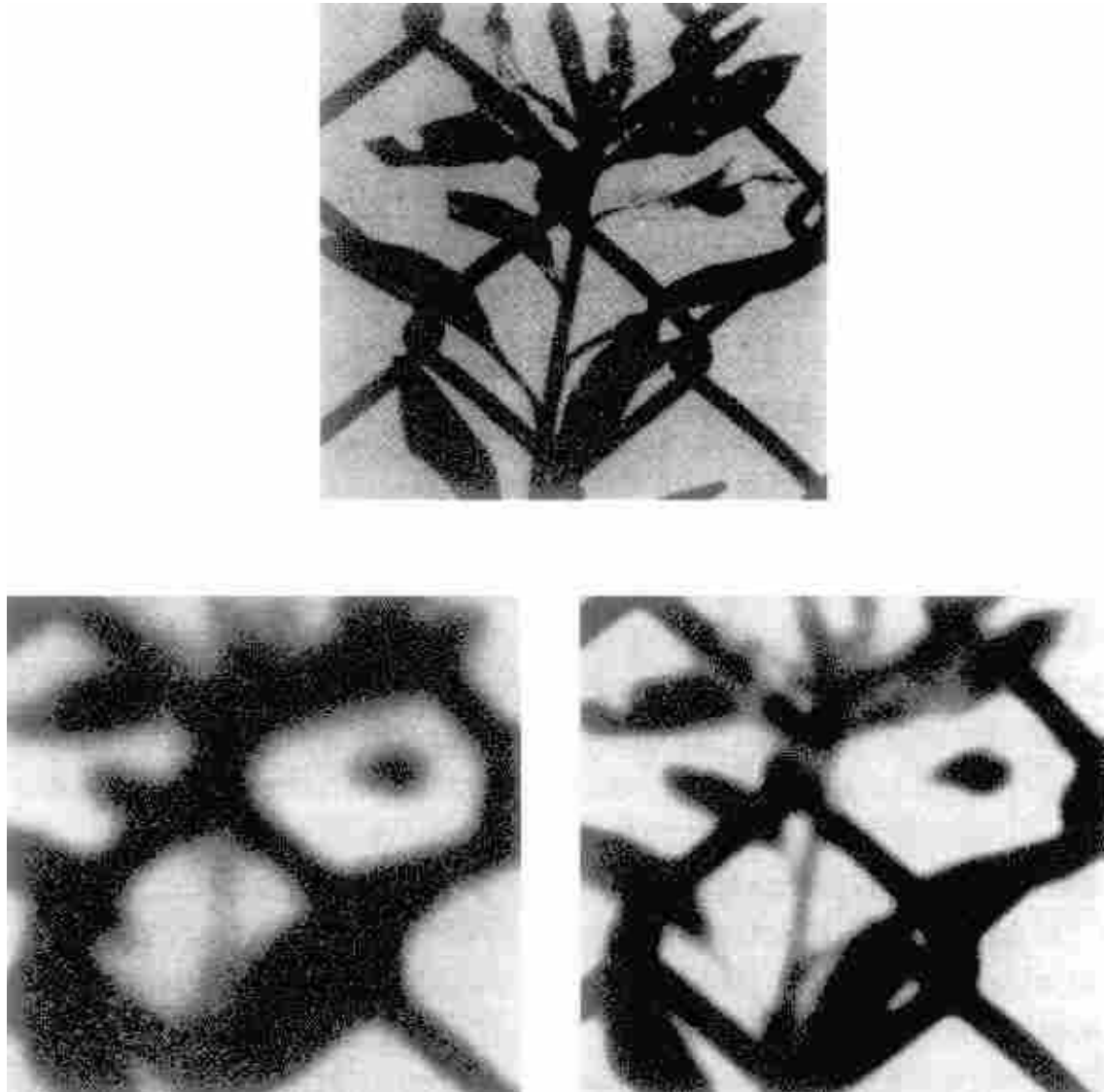


Figura A.1: Imagen original (arriba) hecha borrosa por filtros gaussianos de dos amplitudes diferentes (abajo)

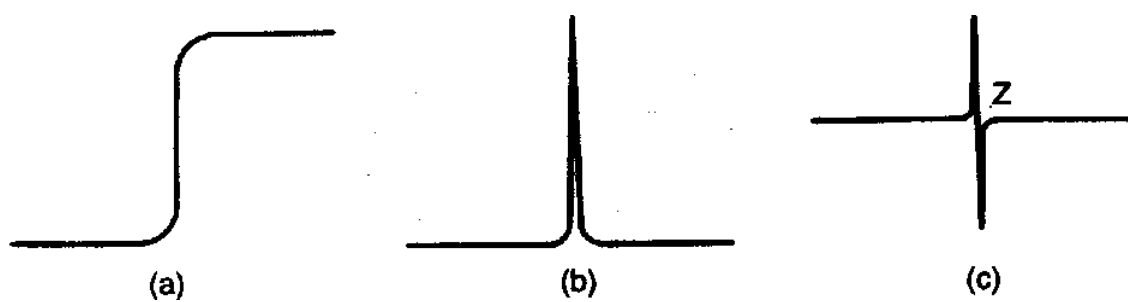


Figura A.2: (a) Intensidad, (b) Derivada Primera (c) Derivada segunda

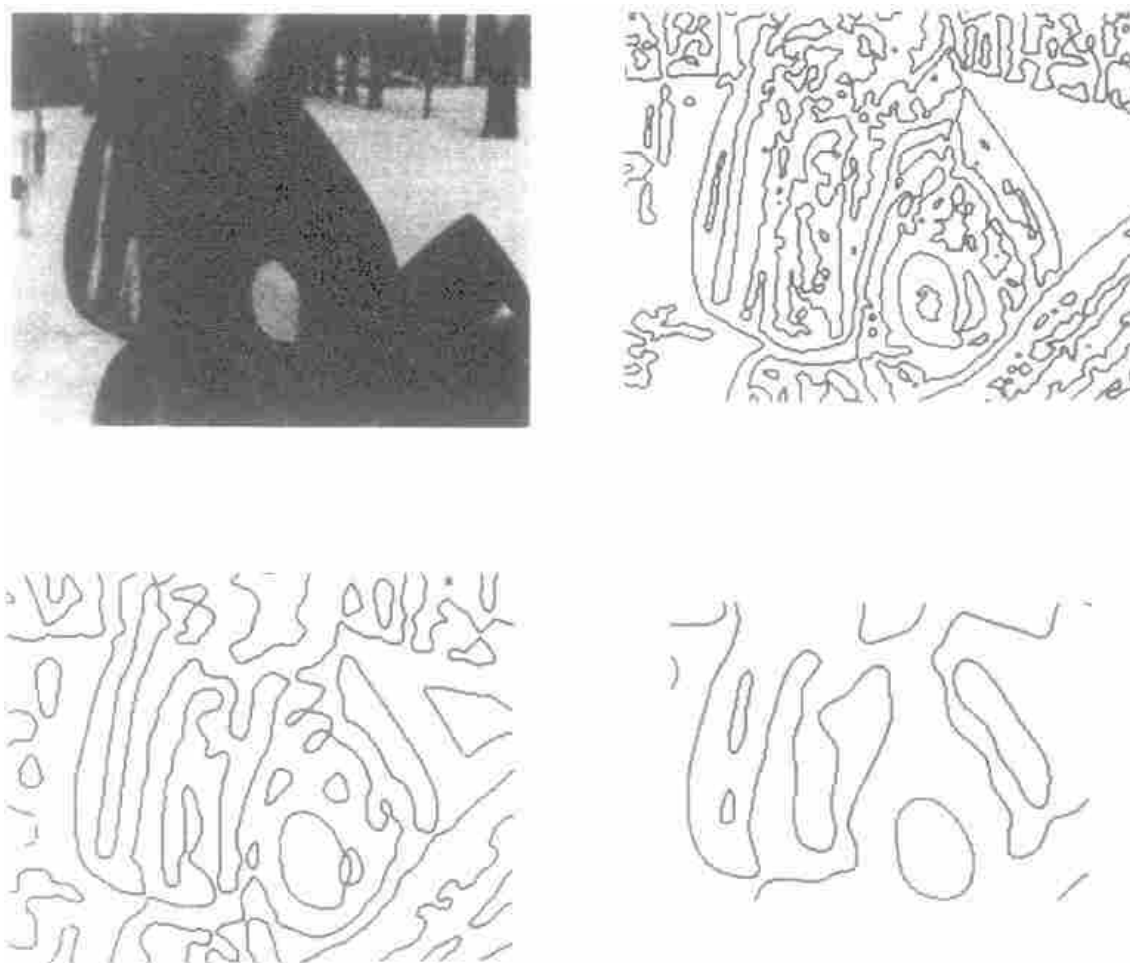


Figura A.3: Resultado de pasar una imagen (superior izquierda) a través de filtros $\nabla^2 G$ con tres amplitudes diferentes, estrecho (superior derecho), intermedio (inferior derecho) y ancho (inferior derecho)

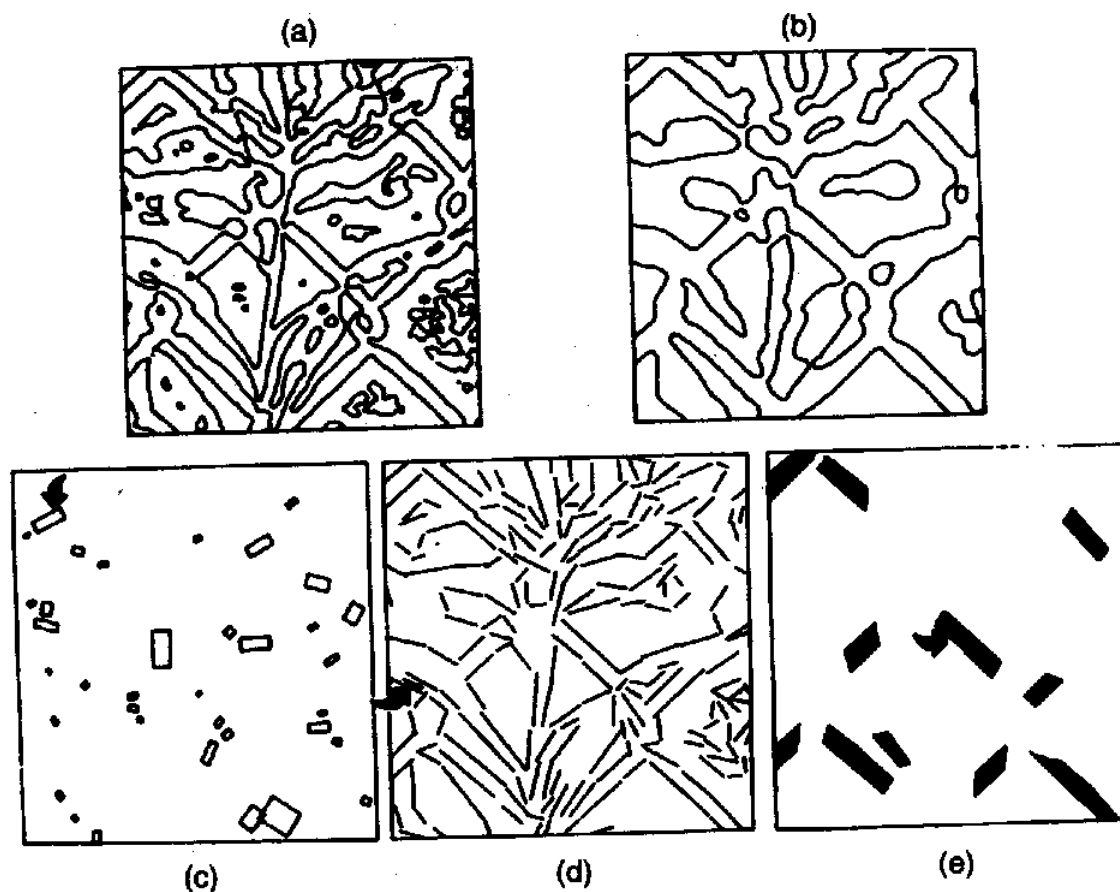


Figura A.4: en (a) y (b) se muestran los pasos por cero de la imagen en figura con un filtro estrecho y otro ancho, cuando se combinan ambas, las localizaciones de las manchas, los segmentos de borde y las barras se muestran en (c), (d) y (e)

mediante *manchas*.

Por tanto, el esbozo primario en bruto es una representación simbólica de la imagen en cuatro señales diferentes: segmentos de borde, barras, terminaciones y manchas, que denotan cuatro tipos diferentes de cambio de intensidad (ver figura A.4).

Marr propuso (ver [28] pg 72-73) que las células de la retina y del córtex visual de los mamíferos funcionaban para localizar pasos de cero en la imagen retiniana, lo que es útil como primer paso hacia la recuperación de la información acerca de los bordes presentes en el mundo. Una comparación de los pasos de cero encontrados por series de células con campos receptivos de diferentes tamaños conduce a una serie de afirmaciones sobre los rasgos presentes en cada posición de la imagen. Este conjunto de declaraciones es el

esbozo primario en bruto.

El algoritmo de Marr y Hildreth trabaja sobre los elementos del esbozo primitivo como sigue, cada uno de los elementos del esbozo tiene asociados atributos de orientación, contraste, longitud, anchura y posición. La representación de una línea recta consistiría en una terminación, después varios segmentos con la misma orientación y a continuación una terminación final. Como ya hemos visto el esbozo primario en bruto es una representación muy compleja y confusa de la que hemos de recuperar las estructuras globales, así como las estructuras internas y la textura superficial.

Esto se logra en el siguiente estadio del procesamiento visual temprano, mediante la asignación recurrente de *indicadores de lugar* a pequeñas estructuras o conjuntos de estructuras, en el esbozo primario en bruto. Estos indicadores de lugar se añaden sucesivamente para formar unidades mayores de un modo cíclico. Los indicadores de lugar pueden definirse mediante la posición de una mancha o de una pequeña línea o un borde; mediante la terminación de un borde o línea más largos, o mancha alargada, o por una pequeña agregación de indicadores. La agregación de estos indicadores de lugar puede proceder por agrupamiento de indicadores próximos basados en cambios de densidad espacial (ver figura A.5), por agregación curvilínea, que produce contornos juntando items alineados que están próximos entre sí (ver figura A.6) y finalmente por la agregación theta. La agregación theta implica el agrupamiento de items orientados similarmente en una dirección que depende, pero difiere, de sus orientaciones intrínsecas (ver figura A.7).

Así, el agrupamiento conjunto de los indicadores de lugar depende de la proximidad local (se combinan elementos adyacentes) y de la similitud (se combinan elementos orientados de modo similar), pero puede también influir en las estructuras detectadas consideraciones más globales. Por ejemplo en la agregación curvilínea, el principio de clausura podría permitir que se unieran dos segmentos borde, aunque el contraste a lo largo de los segmentos borde fuera diferente debido a los efectos de la iluminación (ver figura A.8). Por tanto, el programa de Marr y Hildreth incorpora muchos de los principios de la Gestalt.¹

La aproximación de Marr a la visión enfatiza que debemos considerar lo que los supuestos generales acerca del mundo pueden implicar en el procesamiento visual, para así reducir el margen de posibles interpretaciones de una imagen particular. Los principios

¹Los psicólogos de las Gestalt se identificaban con la muletilla: la totalidad es más que la suma de las partes y una serie de leyes que fueron presentadas como una manifestación de la ley de la pregnancia: de varias organizaciones geométricas posibles, la que de hecho tendrá lugar será aquella que posea la mejor, la más simple y la más estable de las formas.

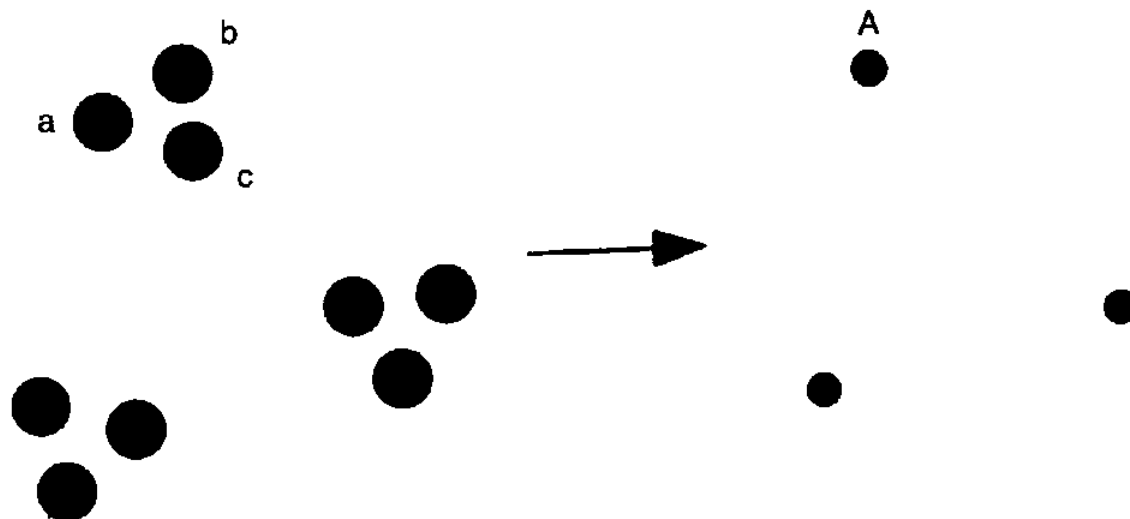


Figura A.5: Agrupación de puntos. a, b y c se agrupan en A

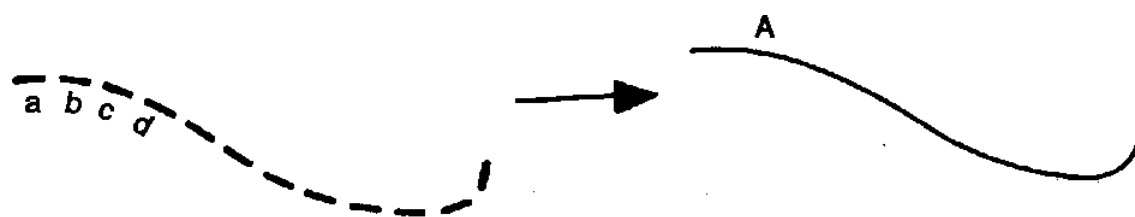


Figura A.6: Agrupación curvilínea. a, b, c, d y sucesivos se agrupan en A

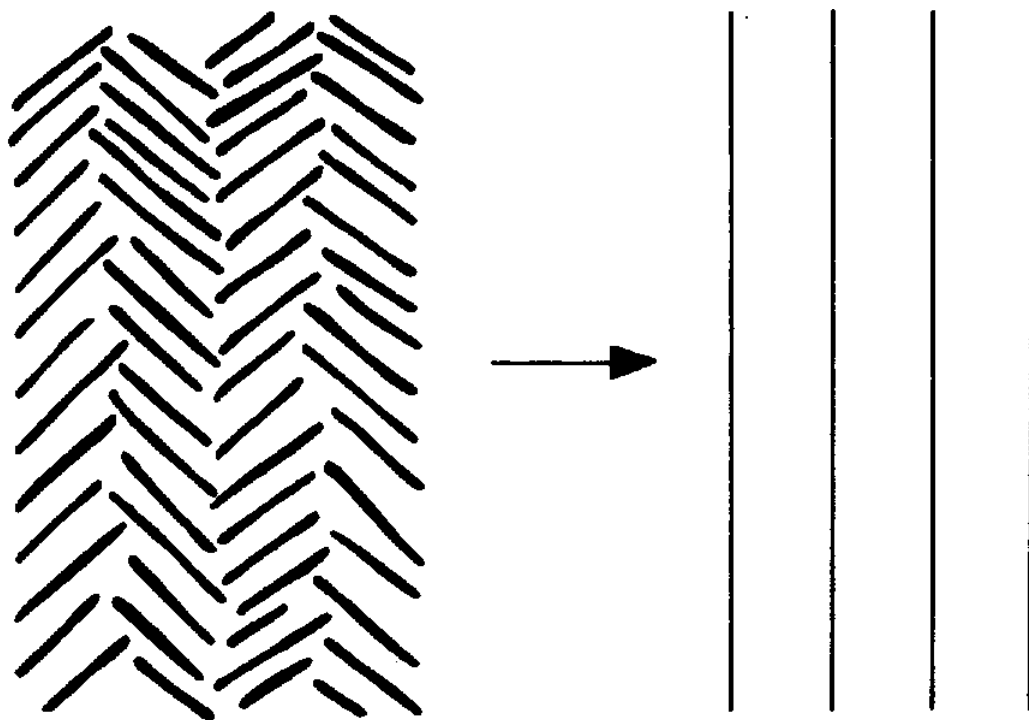


Figura A.7: Ejemplo de agregación theta

de organización de la Gestalt pueden funcionar porque reflejan un conjunto de supuestos razonables que pueden hacerse acerca del mundo de los objetos físicos y biológicos. No obstante, tener una serie de principios descriptivos, aunque sepamos por qué funcionan, todavía es sólo un punto de partida para una teoría completa del procesamiento de información en los procesos de agrupamiento. Necesitamos saber cómo pueden aplicarse tales principios a los elementos primitivos recuperados a partir de las imágenes-bordes, manchas y demás para recuperar las estructuras presentes potencialmente significativas. Es la investigación en Inteligencia Artificial la que ha tratado de proporcionar dicha teoría del proceso de organización perceptual, que es mucho más potente que una teoría simplemente descriptiva, tal como la de los gestaltistas o de investigadores más recientes como Julesz².

²Julesz estudió [23, 24]) el agrupamiento por similitud de modo que incluyera imágenes de textura natural en las que el brillo y el color variasen tanto como la inclinación y la configuración de los elementos, Julesz sugirió que una región de brillo similar o de longitud de onda similar podía agruparse junta, en la medida que fuera distinta respecto de otra región donde el brillo medio o la longitud de onda difiriera de la primera.

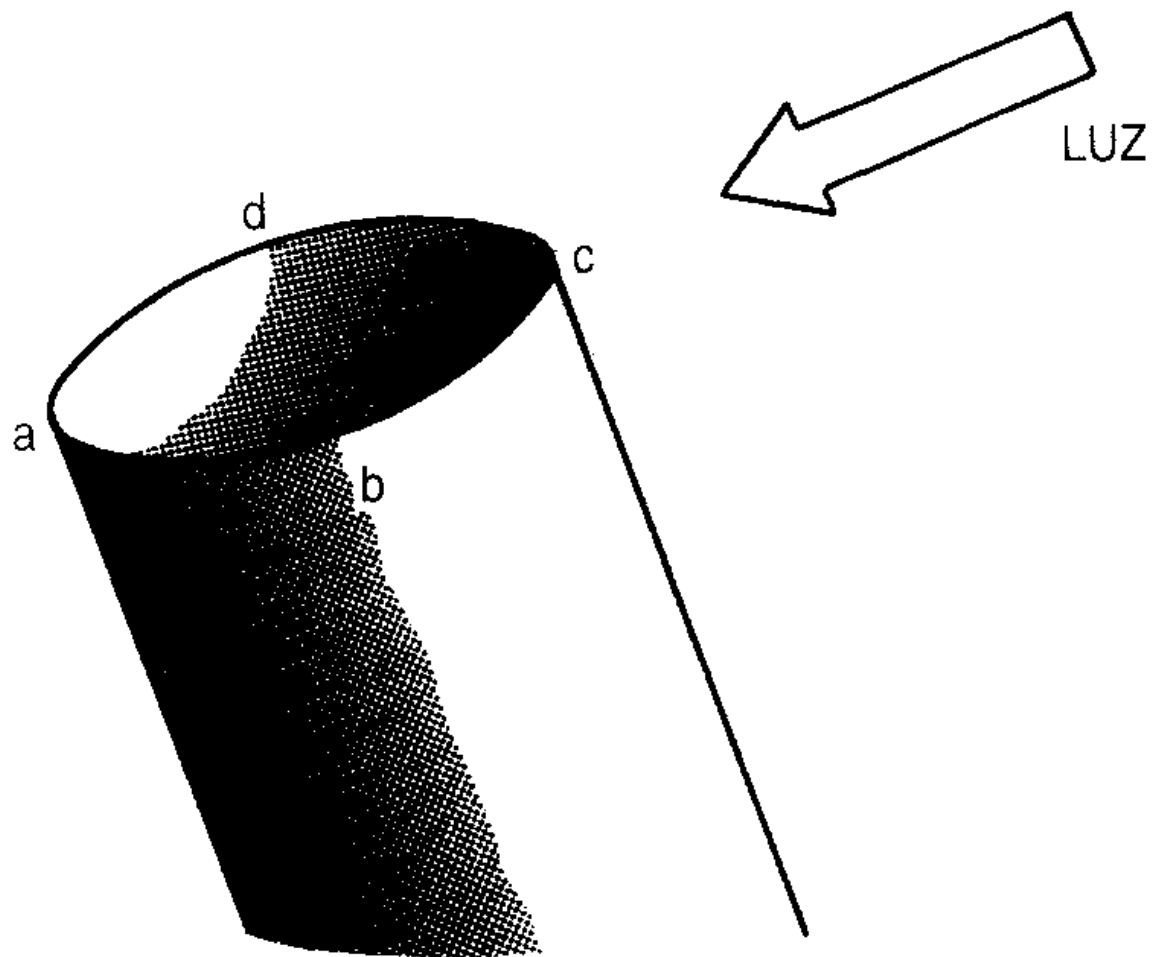


Figura A.8: Agrupación curvilínea junto con clausura puede revelar el contorno a-b-c-d

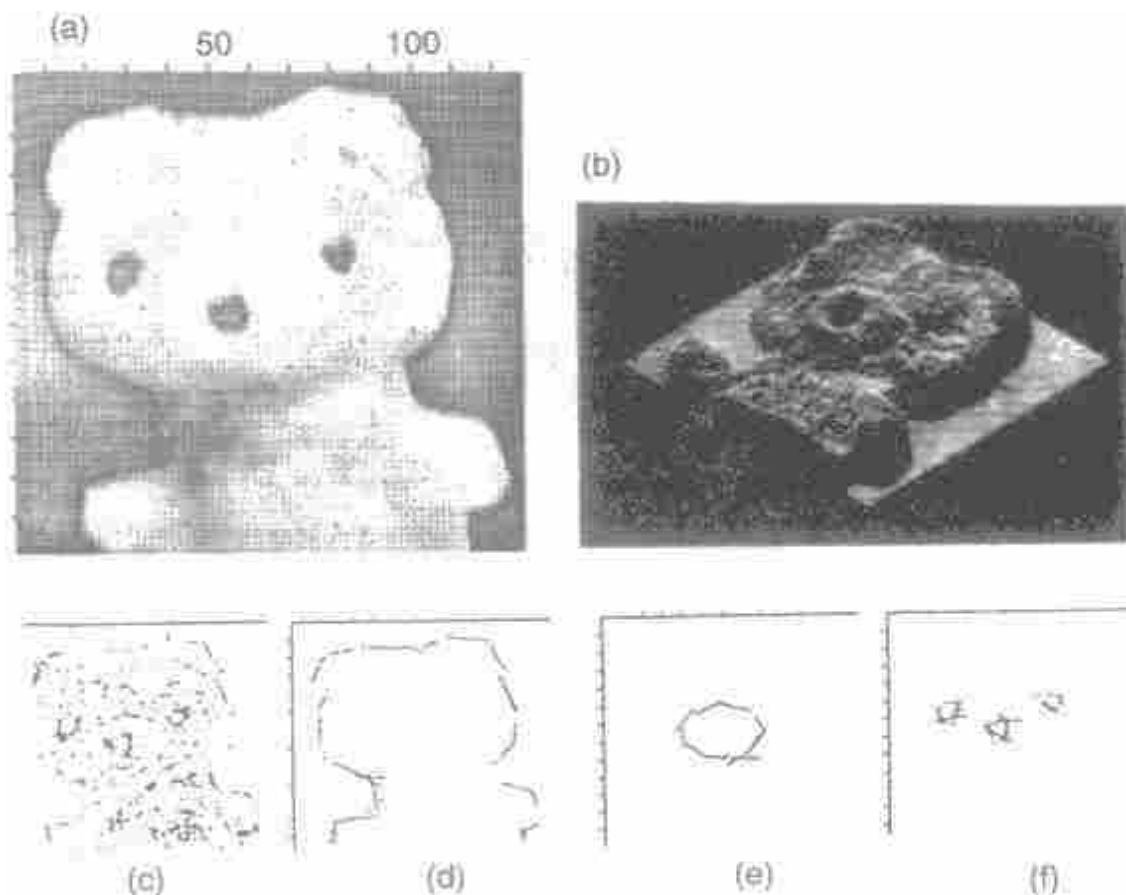


Figura A.9: Osito de peluche, esbozo primario en bruto y agrupamiento

El éxito del programa de procesamiento temprano de Marr puede juzgarse por su habilidad para recuperar los contornos ocultos en la imagen del famoso osito de peluche (ver figura A.9). y revelar los contornos interiores del osito que corresponden a las siluetas de los ojos nariz y hocico. Estas estructuras se recuperan sin recurrir a conocimiento de alto nivel. El programa de Marr trabaja sólo con propiedades generales.

El programa de Marr recurre al conocimiento de alto nivel sólo en caso de ambigüedad, como en el caso de la imagen de la planta (ver figura A.10). Los procedimientos de segmentación fallan al separar los contornos de dos hojas que se superponen. Hace falta que el programa conozca que los diferentes segmentos de contorno no corresponden a una sola estructura. Este conocer del programa se suplirá cuando tengamos acceso a la visión estereoscópica.

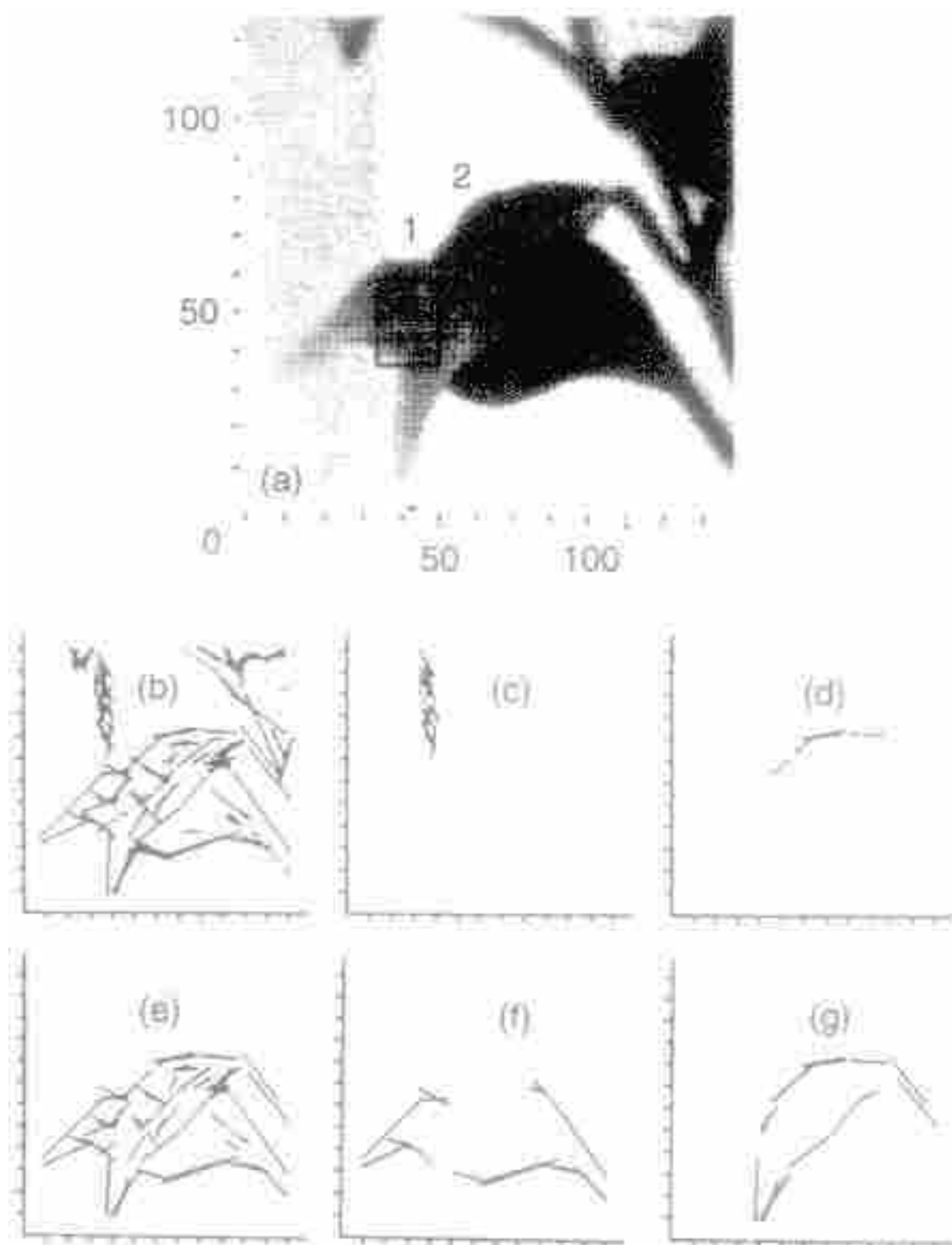


Figura A.10: (a) Hoja, (b) esbozo primario en bruto, (c) y (d) algunas agrupaciones curvilíneas, (e) la separación no es posible (f) y (g) estructuras separadas tras decirle que los elementos de 1 y 2 no van juntos

A.3. Hacia el esbozo $2\frac{1}{2} D$. Percepción de la profundidad y el movimiento

Como ya sabemos Marr veía como objetivo del procesamiento visual temprano la producción de una descripción de las superficies visibles del entorno, de modo que su disposición ordinal y su distribución se describieran con respecto al observador. A esta descripción la llamó esbozo $2\frac{1}{2} D$ y ésta se construye a partir de varias fuentes diferentes. El contorno, la textura y la información de los contornos disponible en el esbozo primario, la estereopsis y el análisis de los movimientos presentes, todo interviene en este esbozo, que actúa como un almacén de memoria en el que las soluciones parciales pueden almacenarse como producto del procesamiento. Analizaremos ahora antes de seguir con el esbozo $2\frac{1}{2} D$, el movimiento y la estereopsis. La percepción de la profundidad y el movimiento exige que se calculen las correspondencias entre las imágenes retinianas desplazadas en el tiempo (de un instante al siguiente) y en el espacio (de cada ojo).

A.3.1. Percepción Estereoscópica

La psicología de la percepción ha estado dominada por la aparente paradoja de la tercera dimensión en visión. Después de todo, aunque curvada la imagen retiniana es esencialmente bidimensional, a pesar de que nuestra percepción sea de un mundo en tres dimensiones ¿Cómo puede recuperarse la profundidad?

En un rango fijo, los animales con campos visuales que se solapan tienen disponible una información estereoscópica a partir de las imágenes dispares obtenidas en los dos ojos. Cada ojo tiene una perspectiva del mundo ligeramente diferente, debido a la separación horizontal de los dos ojos. Si abrimos y cerramos cada ojo alternativamente parecerá que el mundo se desplaza lateralmente.

La figura A.11 muestra las propiedades geométricas de las dos imágenes retinianas que dan lugar a la disparidad. Si enfocamos los dos ojos hacia el objeto A, se dice que las imágenes proyectadas por A en cada ojo se encuentran en puntos correspondientes en las dos retinas. Sin embargo, las imágenes proyectadas por objetos como B incidirán en puntos dispares en las dos retinas, es obvio que la cantidad de disparidad dependerá de la distancia entre A y B. De este modo puede entenderse que si el cerebro es capaz de calcular la disparidad, obtendrá información precisa sobre las distancias relativas de los objetos en el mundo. Es importante hacer notar que se ha observado que la disparidad se

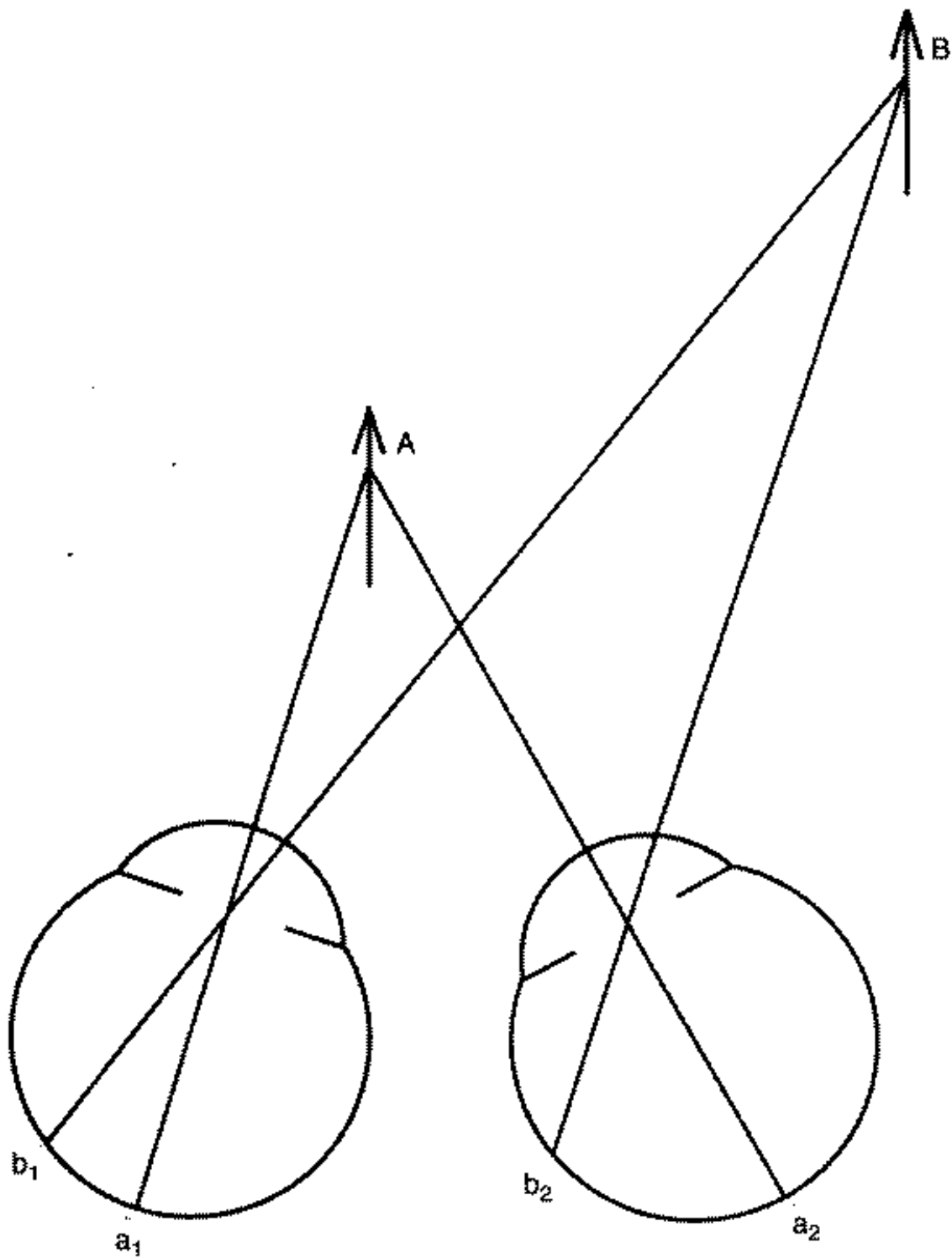


Figura A.11: Los ojos enfocan a A y se proyectan en los puntos de las retinas a_1 y a_2 . B se proyecta en dos puntos más dispares

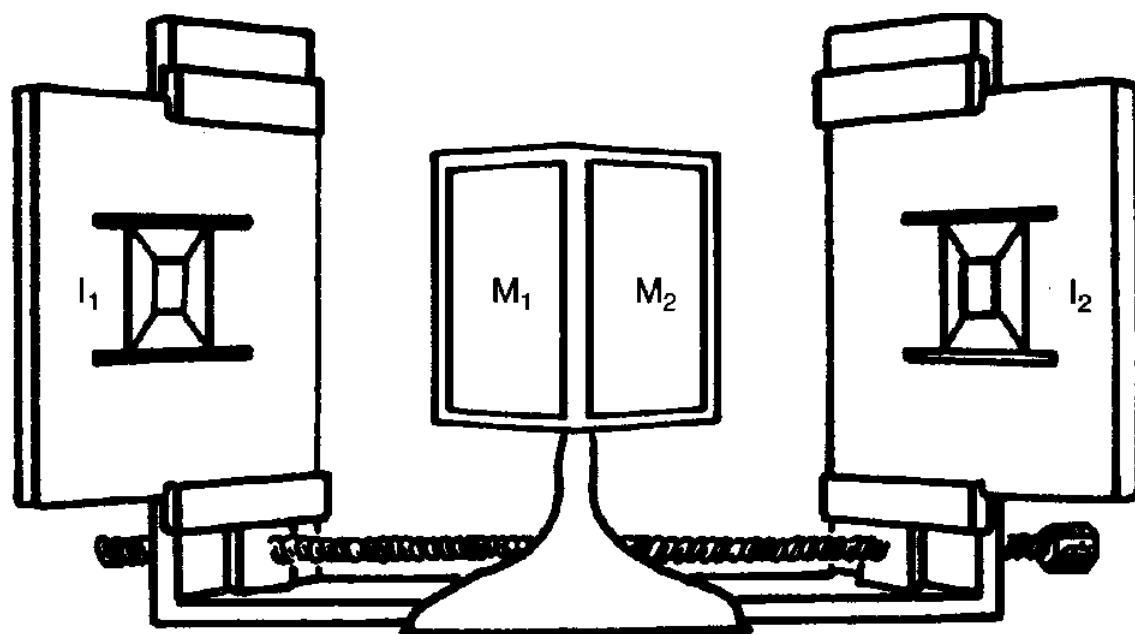


Figura A.12: Estereoscopio de Wheatstone

procesa en la vía visual en una etapa temprana.

Volviendo ahora a la evidencia psicológica, la importancia de la información de disparidad para la percepción de la profundidad puede demostrarse fácilmente. Es posible crear fuertes impresiones de profundidad a partir de dibujos mediante el envío a cada ojo, separadamente, de la proyección que el ojo vería si el objeto, de hecho, se presentara en profundidad. Normalmente, se le atribuye a Wheatstone (1838) la invención del primer estereoscopio, éste se muestra en la figura A.12.

Fusionar imágenes requiere que las disparidades entre las imágenes de los dos ojos se interpreten a partir de objetos individuales con localizaciones únicas en el espacio. Para lograr esta interpretación se requiere a la vez, que decidamos qué partes de la imagen de un ojo corresponden a determinadas partes en la imagen del otro ojo.

Aunque existían investigaciones anteriores, fue Bela Julesz quien en los años sesenta, resumido en su libro de 1971, desarrolló estereogramas (ver figuras A.13 y A.14) de puntos aleatorios como una herramienta para explorar los procesos de estereopsis.

Basado en los resultados obtenidos con estos estereogramas, Julesz propone que la estereopsis se lleve a cabo mediante una comparación punto por punto de los puntos con un mismo valor de brillo. Sin embargo, esto lleva al problema de ambigüedad en el proceso de acoplamiento como muestra la figura A.15. Aunque son varios los posibles acoplamientos

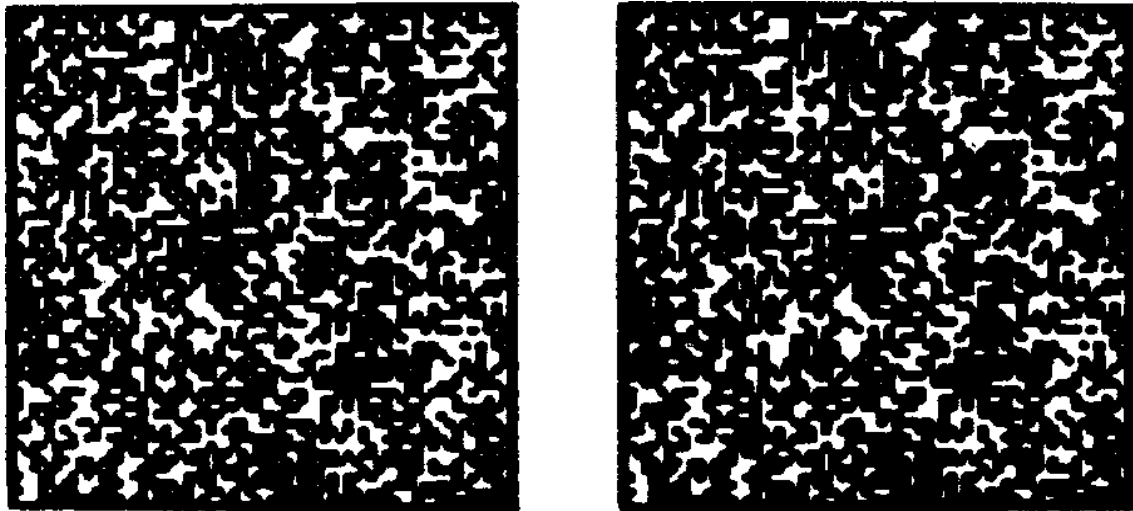


Figura A.13: Estereograma de puntos aleatorios. Si este par estereo fuese visto en un estereoscopio se vería un cuadrado flotando por encima del fondo

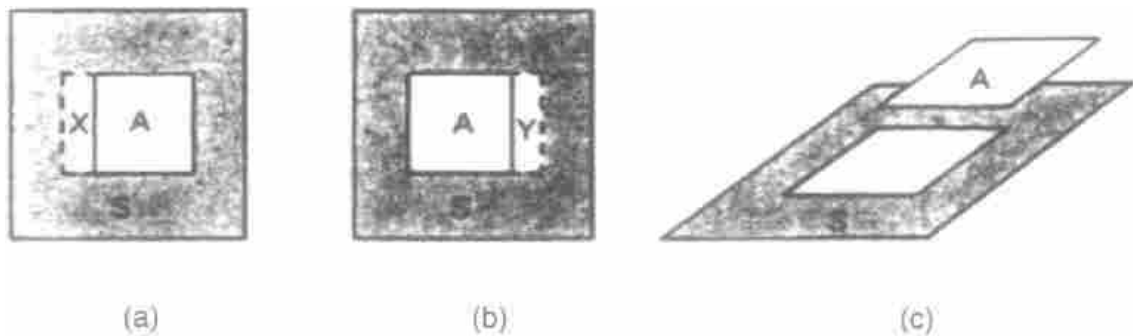


Figura A.14: (a) y (b) son las dos mitades de un estereograma de puntos aleatorios simplificado. Con una zona circundante común (S) y una zona central (A) desplazada hacia la derecha en (a) y la izquierda en (b). El resto se ha llenado con más textura (X e Y). Construyendo tal estereograma se observa un cuadrado central flotando por encima del fondo como se muestra en (c).

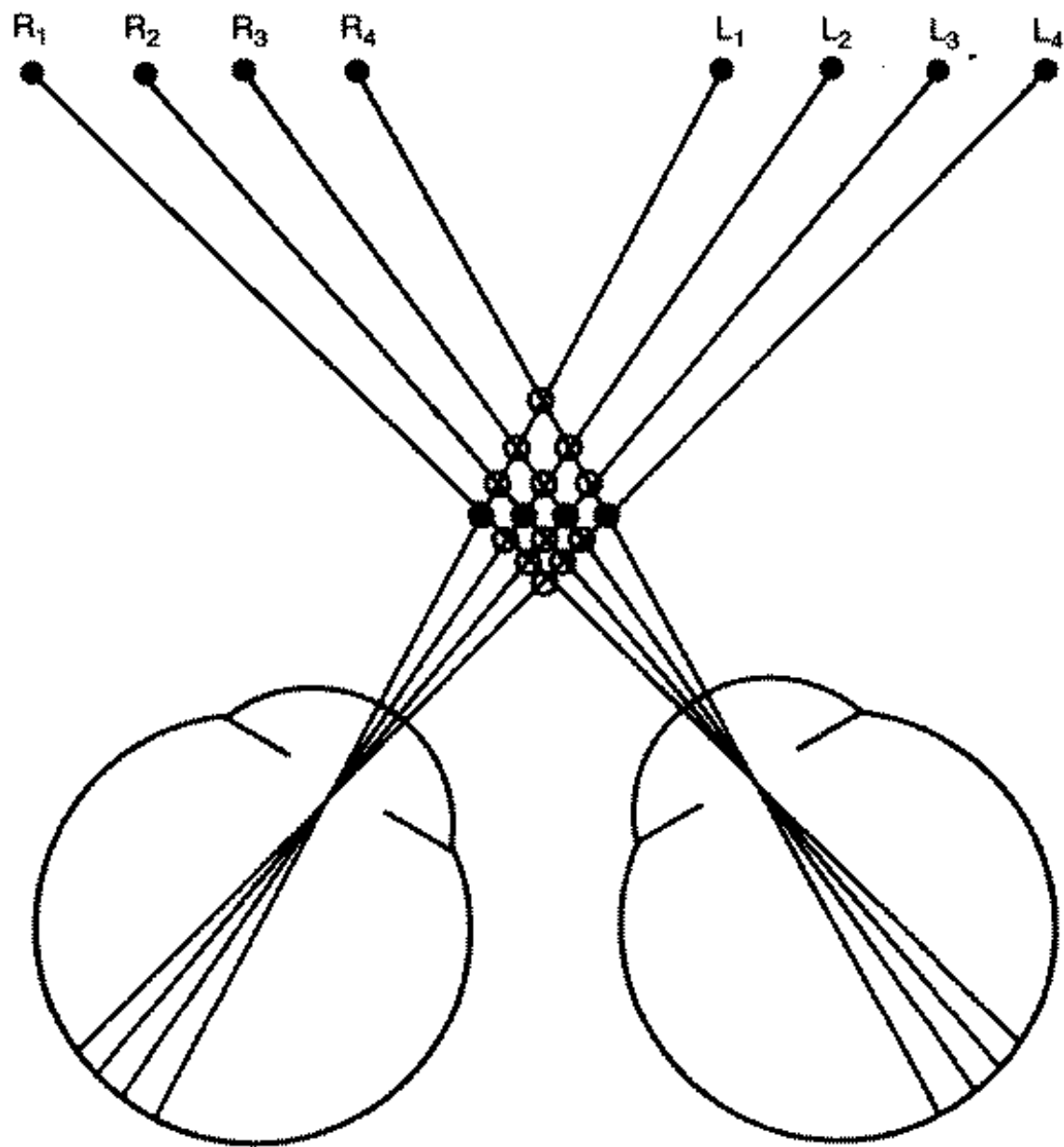


Figura A.15: El acoplamiento que se ve es el mostrado por los puntos negros

sólo persiste uno. Marr y Poggio sugieren entonces como principios generales del mundo físico pueden servir para restringir los acoplamiento que se toleran en la estereopsis. Según los autores para que dos elementos de imagen (uno en cada ojo) correspondan a la misma zona de la superficie deberían tener descripciones físicas similares (restricción de compatibilidad), es decir puntos negros deberían acoplarse con puntos negros. Además cada punto debería tener un único acoplamiento (restricción de unicidad). Finalmente, la restricción de continuidad nos dice que la disparidad debería variar suavemente casi en todos los lugares de la imagen.

Si estas restricciones son aplicadas a los posibles acoplamientos de estereogramas de puntos aleatorios se puede obtener un algoritmo que resuelva este tipo de estereograma ([32]) aplicando las tres reglas de acoplamiento, simultáneamente. Sin embargo, para la estereopsis humana el algoritmo propuesto por Marr y Poggio no es satisfactorio.

Julesz y Miller ([25]) mostraron que el ruido de alta frecuencia espacial añadido a los esterogramas no desbarataba la fusión binocular obtenida con tal de que no hubiera superposición entre las frecuencias espaciales del ruido y la imagen. Esta fue una de las varias demostraciones de que el emparejamiento estereo puede proceder dentro de canales sintonizados a una frecuencia espacial diferente. Esta observación llevó a Marr y Poggio ([33]) a formular un nuevo algoritmo en el que el acoplamiento se realizaba a escalas espaciales cada vez más finas y entre los cruces por cero localizados dentro de canales independientes de $\nabla^2 G * I$. Sin embargo, este algoritmo también tiene sus limitaciones. (ver [5] página 237 y ss, y [20], página 43). Recientemente el algoritmo estereo PMF (ver [5] y [20]) podría ser la clave de la visión humana estéreo.

A.3.2. Percepción del Movimiento

La mayor parte de los animales, incluyéndonos a nosotros mismos, exploran activamente y muestrean sus mundos visuales con los movimientos del ojo, la cabeza y el cuerpo, y los movimientos de las cosas en el mundo pueden indicar sucesos de interés o peligro.

¿Cuáles son las consecuencias del movimiento en el mundo, bien de los ojos o de los patrones de estímulos en la retina?. Es obvio que para este problema hemos de considerar que estamos ante un patrón de intensidades que varía no sólo espacialmente sino también temporalmente. Cualquier movimiento que se produzca por parte del observador, bien en su cuerpo, la cabeza o los ojos producirá una transformación en la imagen retiniana completa, una transformación que recibe el nombre de flujo óptico.

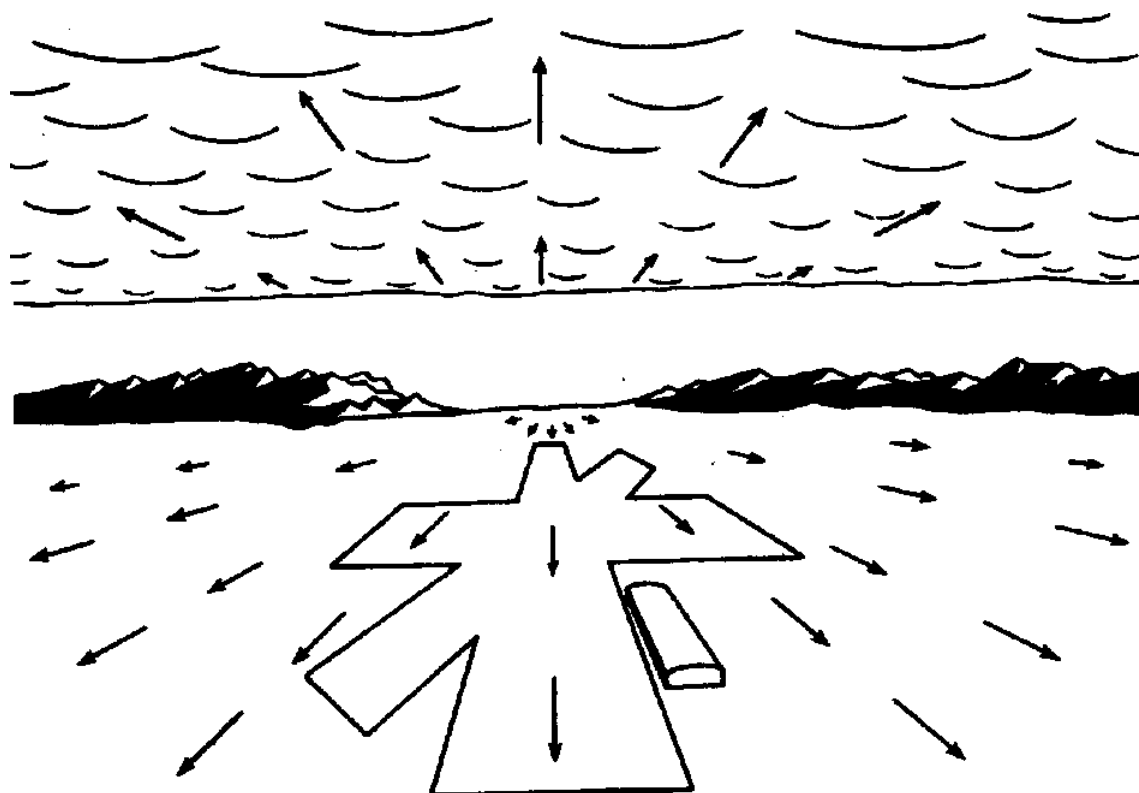


Figura A.16: Campo de flujo óptico para un piloto aterrizando un avión

Si por ejemplo, un observador mueve su cabeza de izquierda a derecha se produce un flujo lateral de información a través de la retina. Otro ejemplo de flujo óptico es el que ve un piloto cuando está aterrizando un avión (ver figura A.16). El flujo óptico que resulta de nuestros movimientos nos informa de nuestros propios movimientos. Esto se demuestra informalmente en situaciones donde el flujo óptico sólo produce la falsa impresión de que nos estamos moviendo, un ejemplo se produce cuando creemos que nuestro propio tren sale de la estación cuando sólo el tren vecino está entrando. Además de informarnos sobre nuestros propios movimientos hay evidencia de que los observadores humanos usan los patrones en expansión del flujo óptico para informarnos sobre el tiempo para contactar con una superficie hacia la que nos dirigimos y por tanto para recuperar información sobre la distancia a dicha superficie.

En contraste con los patrones globales de flujo óptico producidos por el movimiento del observador, el movimiento de un objeto en el mundo producirá una distorsión en el patrón en la retina. Estos patrones de distorsión local pueden informarnos a su vez de las estructura tridimensional en movimiento. Tal vez el ejemplo más famoso es el de

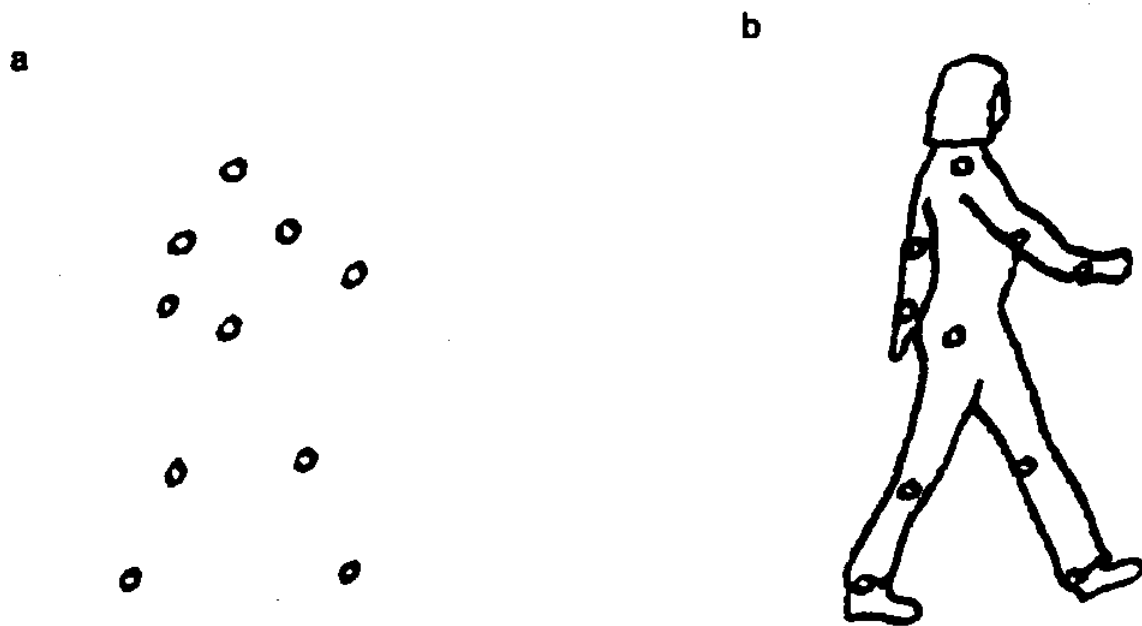


Figura A.17: La demostración de los puntos iluminados

figuras humanas proporcionado por la demostración de Johansson. Johansson (ver [22]) iluminó puntos de articulaciones (hombros, codos, rodillas, etc) de actores y los filmó en la oscuridad de forma que sólo esos puntos fueran visibles (figura A.17). Cuando el personaje no se movía el objeto no tenía significado, pero tan pronto como comenzaba el movimiento, se podían ver las formas humanas, y los movimientos ser descritos precisamente. Es obvio por tanto que el patrón completo de flujo debe ser descompuesto en diferentes componentes de traslación y rotación, para que por una parte informe al observador de la naturaleza de sus movimientos en el mundo y para revelar los patrones residuales de movimiento atribuibles a sucesos en el mundo que son externos a los movimientos del observador.

¿Cómo ordena el sistema visual todas estas diferentes componentes de movimiento?
 ¿Cómo se relaciona el análisis del movimiento con el análisis de las estructuras en el mundo?.

La persistencia y utilidad del movimiento visual constituye una de las mayores influencias en el desarrollo de la teoría de la *percepción directa* de Gibson ([11]), en esta teoría el autor niega que la percepción visual lleve asociada inferencia a partir de imágenes retinianas instantáneas. En su lugar Gibson defiende que la información para especificar las estructuras en el mundo y la naturaleza de los movimientos de un observador se detectan a partir de los patrones dinámicos de flujo óptico.

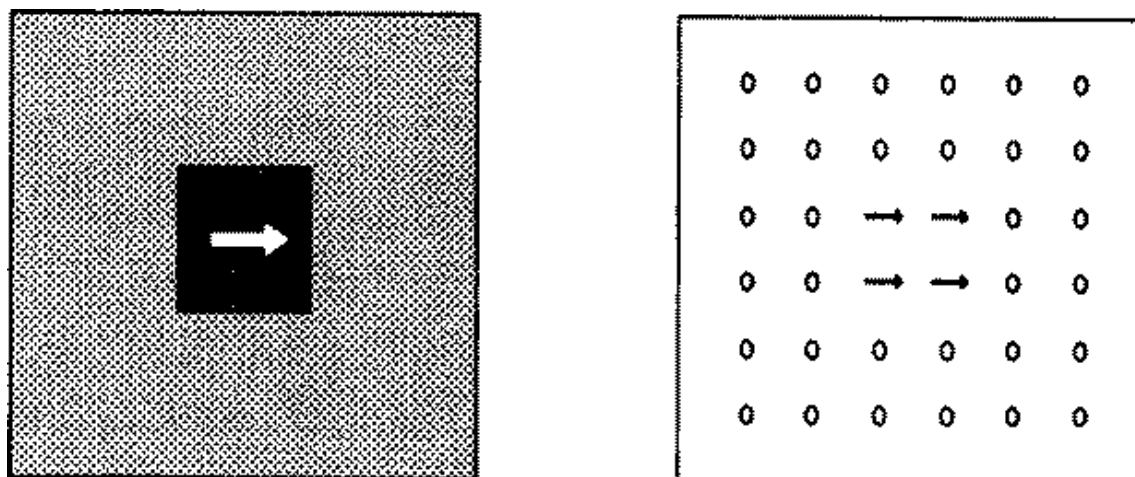


Figura A.18: (a) Una imagen consistente en un cuadro que se mueve contra un fondo estacionario. (b) Su representación en el campo de velocidad, especificando la velocidad y dirección del movimiento en cada región de la imagen

Aunque se podría estar de acuerdo con Gibson en que las transformaciones en la estructura de la luz que llega a un observador son una fuente esencial de información, sin embargo es posible rechazar una teoría de la percepción directa. Las evidencias contribuyen a una teoría computacional de la información disponible en el flujo óptico. Para comprender cómo se obtiene esta información, necesitamos elaborar algoritmos capaces de extraerla a partir de una imagen retiniana que varía con el tiempo y, luego, preguntarnos si estos algoritmos se encuentran implementados en los sistemas nerviosos.

Podemos definir la etapa temprana de procesamiento de movimiento como aquella que tiene como entrada una matriz, que varía con el tiempo, de valores de intensidades de grises, y que produce como salida el campo de la velocidad; una matriz de medidas de velocidad de las imágenes en cada pequeña región de la imagen, las cuales pueden a la vez utilizarse como entradas de otros procesos que obtengan propiedades de alto nivel del movimiento de la imagen. Puesto que la velocidad es un vector, cada medida debe especificar tanto la dirección como la magnitud del movimiento. (ver figura A.18).

Analicemos ahora el procesamiento temprano del movimiento según el modelo de Marr. El objetivo del procesamiento temprano del movimiento es obtener los valores de la velocidad a partir de los valores de intensidad variables en el tiempo. ¿Cómo se puede conseguir esto?. Primero, hay que tener en cuenta que la velocidad no se puede determinar a partir de los cambios de intensidad en un solo punto. La figura A.19 nos muestra

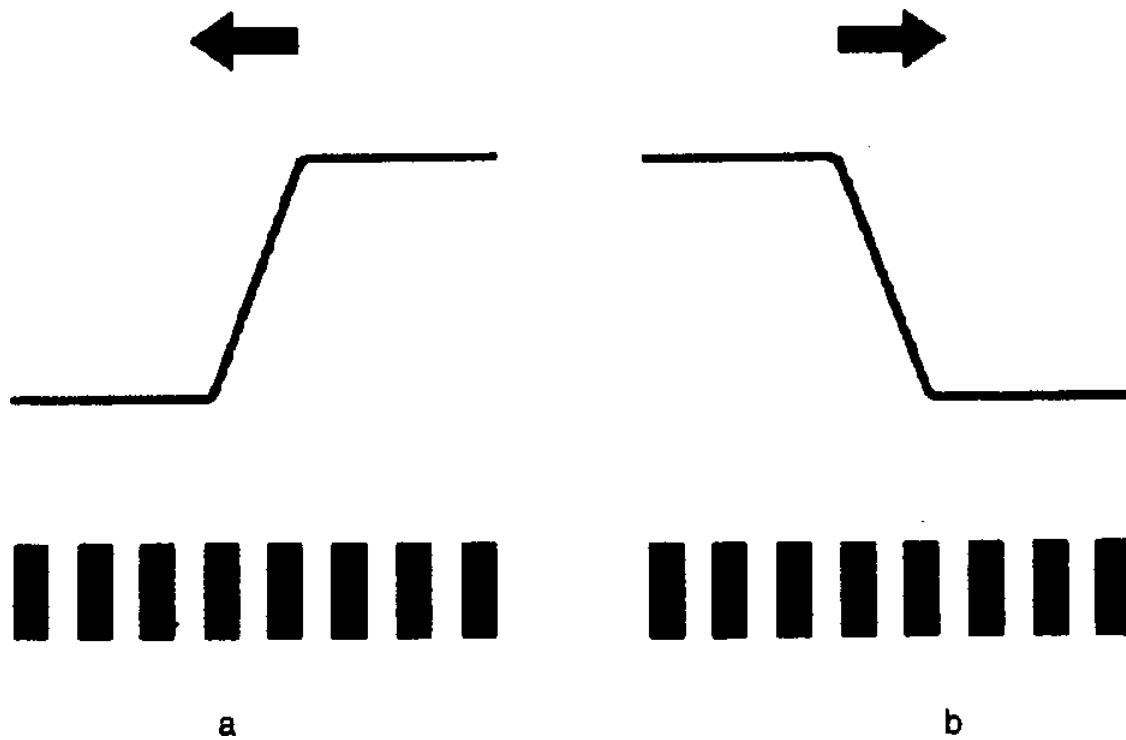


Figura A.19: Gradientes de intensidad de luz pasando sobre una fila de receptores

dos gradientes de intensidad que se mueven sobre una fila de receptores. Conforme pasa sobre cada uno de éstos, la intensidad de la luz aumenta. Sin embargo, este cambio no nos proporciona información sobre la dirección del movimiento (dos bordes moviéndose en dirección opuesta producen el mismo efecto). Tampoco tenemos información sobre la velocidad de movimiento del borde: una determinada tasa de cambio de intensidad podría venir provocada por un borde fino que se mueve lentamente o por uno difuso moviéndose rápidamente. Naturalmente, las medidas de velocidad deben obtenerse por medio de algún cálculo realizado sobre valores de intensidad en un número de receptores contiguos.

Como ya hemos dicho antes, la velocidad de cambio de intensidad de luz en un receptor no puede proporcionar información sobre la velocidad o dirección del movimiento de la imagen. La ambigüedad en la dirección del movimiento se podría resolver si se procesara el signo del gradiente espacial de intensidad de luz; por ejemplo, un incremento en la intensidad junto con un gradiente espacial positivo indica, sin ambigüedad, movimiento hacia la izquierda. Además, si se conoce la pendiente del gradiente espacial, también se puede determinar la velocidad del movimiento de la imagen. La relación entre la velocidad, la tasa de cambio de la intensidad en un punto y el gradiente espacial de intensidad en

torno a ese punto vienen dados por

$$v = -\frac{\delta I}{\delta t} \frac{\delta x}{\delta I}$$

donde v es la velocidad de la imagen y $\delta I/\delta t$ y $\delta x/\delta I$ son las derivadas parciales de la intensidad respecto al tiempo y de la distancia a lo largo del eje x . Si se pueden calcular estas dos derivadas, será posible medir sin ambigüedad, la velocidad a lo largo de un único eje. La derivada del tiempo podría obtenerse a partir del cambio de intensidad de un único receptor, pero el cálculo del gradiente espacial requiere de dos o más receptores vecinos.

Una pregunta obvia es si este tipo de algoritmo se halla implementado en cualquier sistema visual. Marr y Ullman ([31]) sugieren que sí lo está y en particular ya que se miden los gradientes en intensidad mediante el algoritmo $\Delta^2 G$ de Marr y Hildreth, la velocidad del movimiento de estos gradientes se procesa al mismo tiempo. La presencia de un gradiente espacial viene indicado por un paso de cero en la salida de un filtro $\nabla^2 G$ y Marr y Ullman proponen que se halle la diferencial con respecto al tiempo para producir una medida de la velocidad.

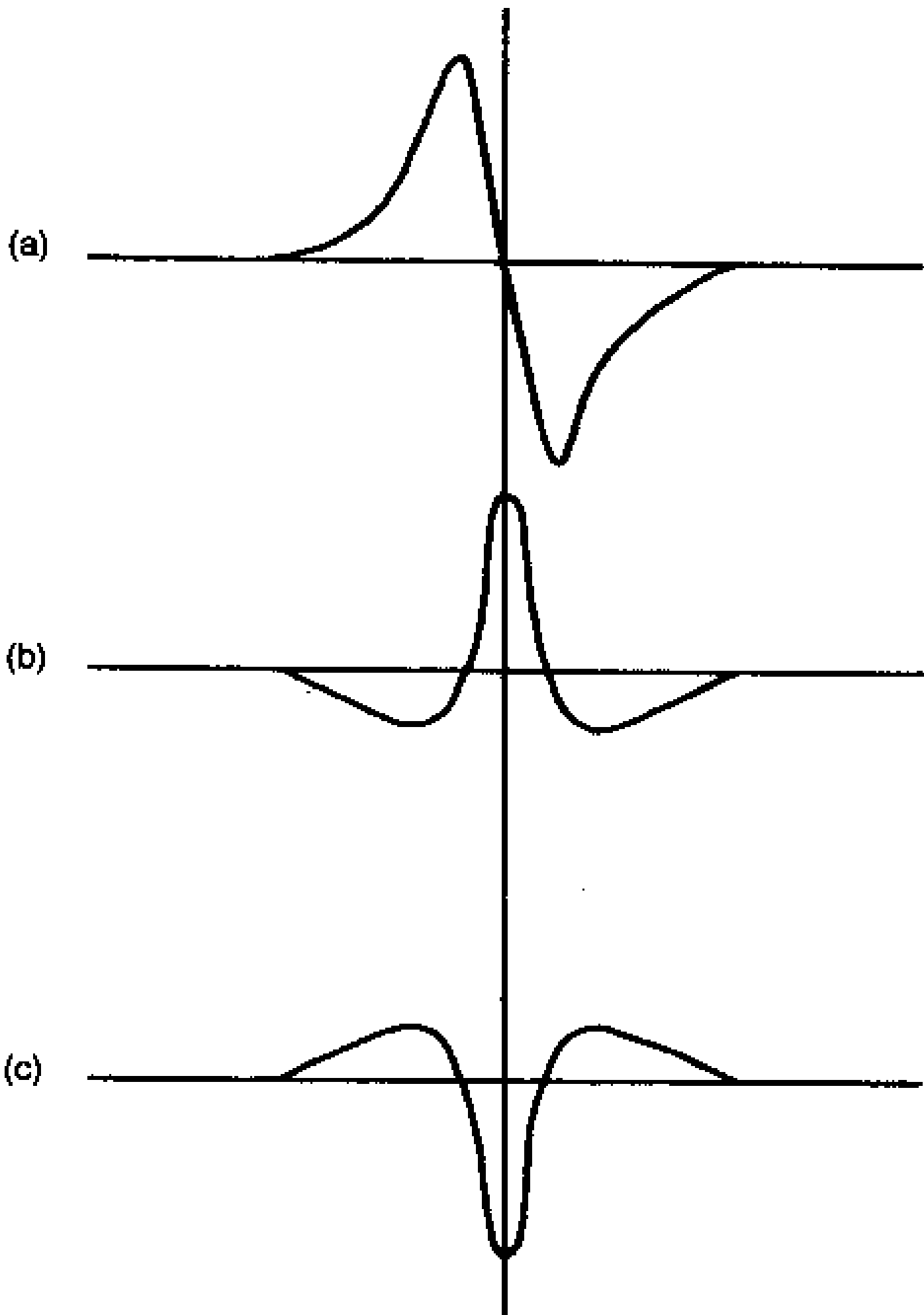
La figura A.20a muestra el perfil de los valores de $\nabla^2 G \times I$ alrededor del paso de cero en X ; si todo el patrón de luz se mueve hacia la derecha los valores de $d(\nabla^2 G * I)/dt$ son como se muestra en la figura A.20b, mientras que si se mueven hacia la izquierda son como se muestra en la figura A.20c. El movimiento hacia la derecha produce un valor positivo en la derivada del tiempo en el punto X , mientras que el movimiento hacia la izquierda produce un valor negativo. Por lo tanto el algoritmo puede procesar la dirección del movimiento a lo largo de un eje. Sobre el procesamiento en la retina de estas derivadas y el uso de las células ganglionares puede verse [5] página 445.

A.3.3. Sobre la percepción de otros aspectos de la visión

Aunque en las subsecciones anteriores hemos estudiados con cierto detenimiento el modelo propuesto por Marr para algunos procesos de la visión, el modelo de Marr es mucho más completo e incluye además los módulos siguientes forma, textura, sombra, color, etc. El lector interesado debe consultar el capítulo tercero de su famoso libro.

A.3.4. Recapitulación sobre la teoría del esbozo $2\frac{1}{2} D$ de Marr

Las ideas que hemos presentado sobre el enfoque de la percepción de la profundidad y del movimiento se incorporan, como ya sabemos, en un solo tratamiento teórico por



Marr([28]). Como ya sabemos Marr ve el objetivo del procesamiento visual temprano como la producción de una descripción de las superficies visibles del entorno, de modo que su disposición ordinal y su distribución se describan con respecto al observador. A esta descripción la denomina esbozo $2\frac{1}{2}D$ y se construye a partir de varias fuentes diferentes. El contorno, la textura y la información del sombreado disponibles en el esbozo primario completo, la estereopsis y el análisis de los movimientos presentes, todo interviene en el esbozo $2\frac{1}{2}D$, que actúa como un almacén de memoria en el que las soluciones parciales pueden almacenarse como producto del procesamiento. La etiqueta 21/2D deriva del supuesto de que el esbozo capta una enorme cantidad de profundidades relativas y orientaciones de superficie, cambios locales y discontinuidades es éstas. Sin embargo, algunos aspectos están representados más exactamente que otros. En particular, Marr concluye que el esbozo $2\frac{1}{2}D$ representa la orientación de la superficie mucho más exactamente que la profundidad. (ver figura A.21) Las discontinuidades en profundidad vienen señaladas, sobre todo por los mecanismos de estereopsis y por la presencia de oclusión. La oclusión puede especificarse por la presencia de contornos oclusivos en el esbozo primario y por discontinuidades en el patrón de movimiento presente. La información sobre la orientación de la superficie viene proporcionada por la estereopsis, por los contornos de superficie y de textura, y por un análisis de la estructura a partir del movimiento.

Finalmente, una nota sobre la modularidad. Marr concibió la organización del sistema visual como intrínsecamente modular, de modo que la estereopsis, la estructura a partir del movimiento, la figura a partir del sombreado y así sucesivamente pudieran estudiarse todas en relativo aislamiento, dado que son procesos que actúan en gran medida uno independiente de otro. Diferentes teóricos sugieren que es el acto de reconocimiento de un determinado objeto, en una determinada localización el que junta las descripciones separadas.

Marr considera el esbozo $2\frac{1}{2}D$ como el producto final del procesamiento visual temprano. Aunque dicho esbozo todavía no ha sido implementado en un ordenador, sugiere que puede escribirse en gran medida sin recurrir a información que fluya hacia abajo. Sin embargo, es obvio que nos queda todavía proceso de reconocimiento.

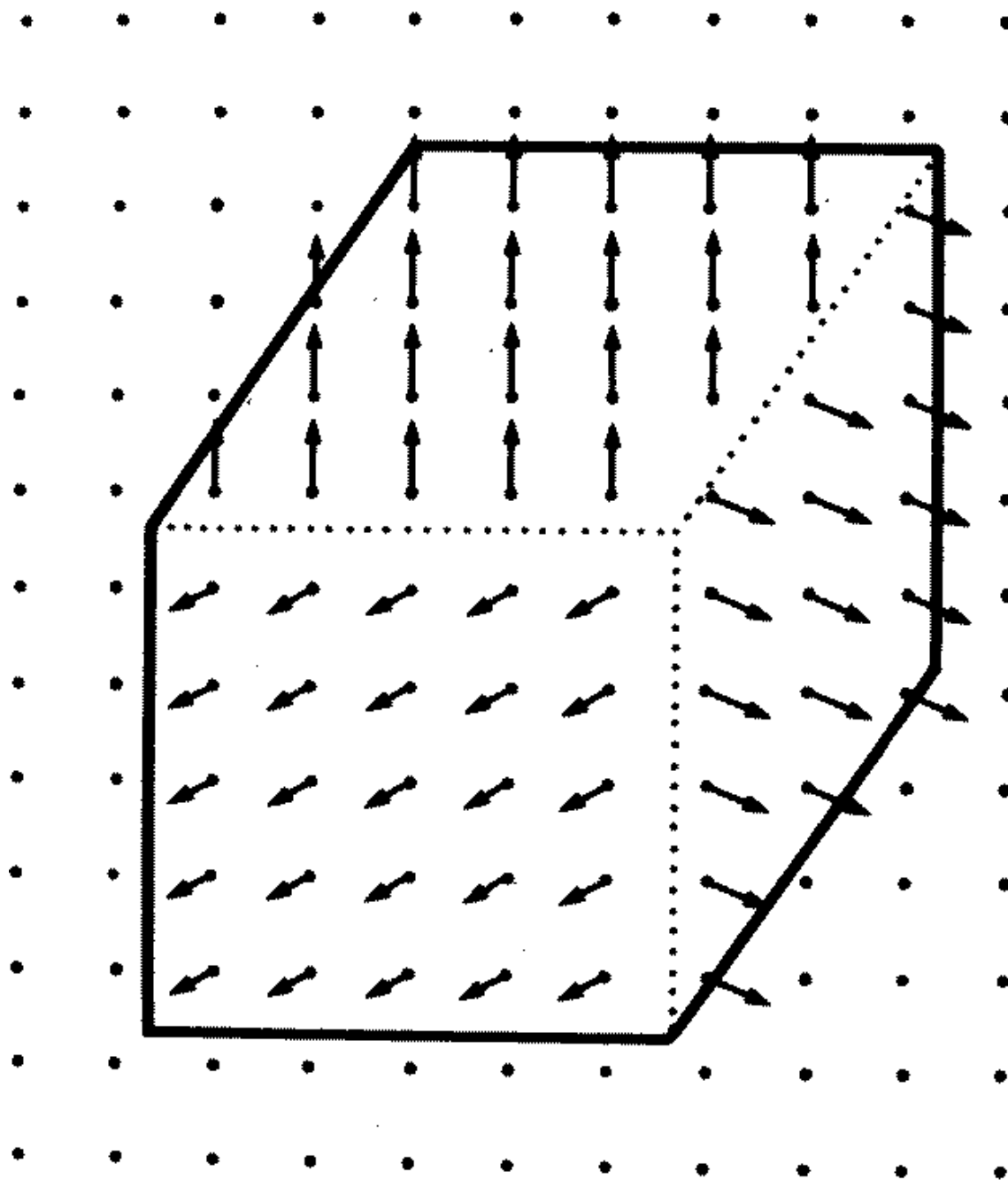


Figura A.21: Esbozo $2\frac{1}{2}D$ de un cubo. Las superficies del cubo están representadas por una serie de vectores primitivos como agujas. La longitud representa el grado de inclinación de la superficie, y la orientación de la aguja, la dirección en la que se inclina.



Figura A.22: diferentes formas de la letra A

A.4. Teoría del Reconocimiento del Objeto de Marr y Nishihara

Una parte esencial de la conducta de los animales y las personas es su habilidad para el reconocimiento de los objetos, animales y personas que son importantes para su supervivencia. Las personas son capaces de reconocer a un gran número de otras personas, las letras del alfabeto, edificios familiares, etc. Los animales necesitan reconocer puntos destacados del terreno, presas apropiadas, parejas o depredadores potenciales y comportarse de la manera apropiada ante cada categoría.

Si damos por sentado que la información disponible para una persona o animal es una imagen bidimensional estática en la retina, inmediatamente surge el problema para explicar el reconocimiento visual. Tomemos el ejemplo de una persona que reconoce las letras del alfabeto: el problema radica en que a una determinada letra le puede corresponder un número infinito de imágenes retinianas, dependiendo de como ha sido escrita la letra, cómo es de grande, ángulo desde el que la vemos, etc (ver figura A.22), aunque de alguna manera reconocemos todos estos patrones de luz como correspondientes a la misma letra. Consideremos ahora el problema del reconocimiento de la cara de un amigo: la imagen de su cara en la retina dependerá de las condiciones de iluminación y de su distancia, ángulo y expresión facial. Otra vez, todas estas imágenes se clasifican conjuntamente, aunque algunas son bastante desiguales.

Ambos casos ilustran el problema de la equivalencia del estímulo, si el que controla la

experiencia es un patrón de luz, o una imagen, sobre la retina entonces hay un número infinito de imágenes que son equivalentes en sus efectos y diferentes respecto a otros conjuntos de imágenes. Obviamente, todas las imágenes que corresponden a una cosa en concreto, ya sea una letra del alfabeto o una cara, deben de tener algo en común, pero el problema es encontrar exactamente esto y como se detecta. Analizaremos aquí el modelo propuesto por Marr, el lector interesado en otros modelos puede consultar, entre otros, el libro [5].

Marr y Nishihara ([30]) sentaron los fundamentos para una posible teoría del reconocimiento. Un objeto debe describirse dentro de un marco de referencia basado en la forma del mismo. Para realizar esto debemos ser capaces de establecer un marco de coordenadas canónicas (un marco de coordenadas que venga determinado por su forma) para la forma antes que ella haya sido descrita.

La serie adecuada de elementos descriptivos (primitivos) para describir una forma dependerá, en parte, del nivel de detalle que la forma ha de captar. Los dedos de una mano no están expresados en un sistema que utilice como primitivas el tamaño de los brazos y las piernas. Para empezar a abordar este problema Marr y Nishihara sugieren que necesitamos una organización modular de las descripciones de la forma con primitivos de diferentes tamaños utilizados a diferentes niveles. Esto permite que una descripción de alto nivel sea estable por encima de los cambios en el detalle fino, pero la sensibilidad a dichos cambios debe estar disponible en otros niveles.

Primero hemos de definir un eje para la representación de la forma. Las formas que son alargadas o tienen un eje de simetría son las más fáciles de describir y Marr y Nishihara limitan su tratamiento a la clase de tales objetos, que pueden describirse como una serie de uno o más conos generalizados. Un cono generalizado es una superficie creada al mover a lo largo de un eje una sección transversal de forma constante, pero de tamaño variable (ver figura A.23). La sección transversal puede hacerse más gruesa o más delgada con tal de que se conserve su forma. Una descripción que utilice primitivos basados en ejes es como una figura hecha de palos. Estas figuras captan las longitudes relativas y la disposición de los ejes que forman los componentes de la estructura completa. Los grosores relativos podrían incluirse también. Podemos usar un solo palo para representar una pierna completa, o bien tres palos más pequeños para representar los segmentos de dicho miembro, el superior, el inferior y el pie. Obviamente podemos ir a niveles más finos. En cada nivel de la descripción podemos construir un modelo 3D, en el que cada modelo especifica:

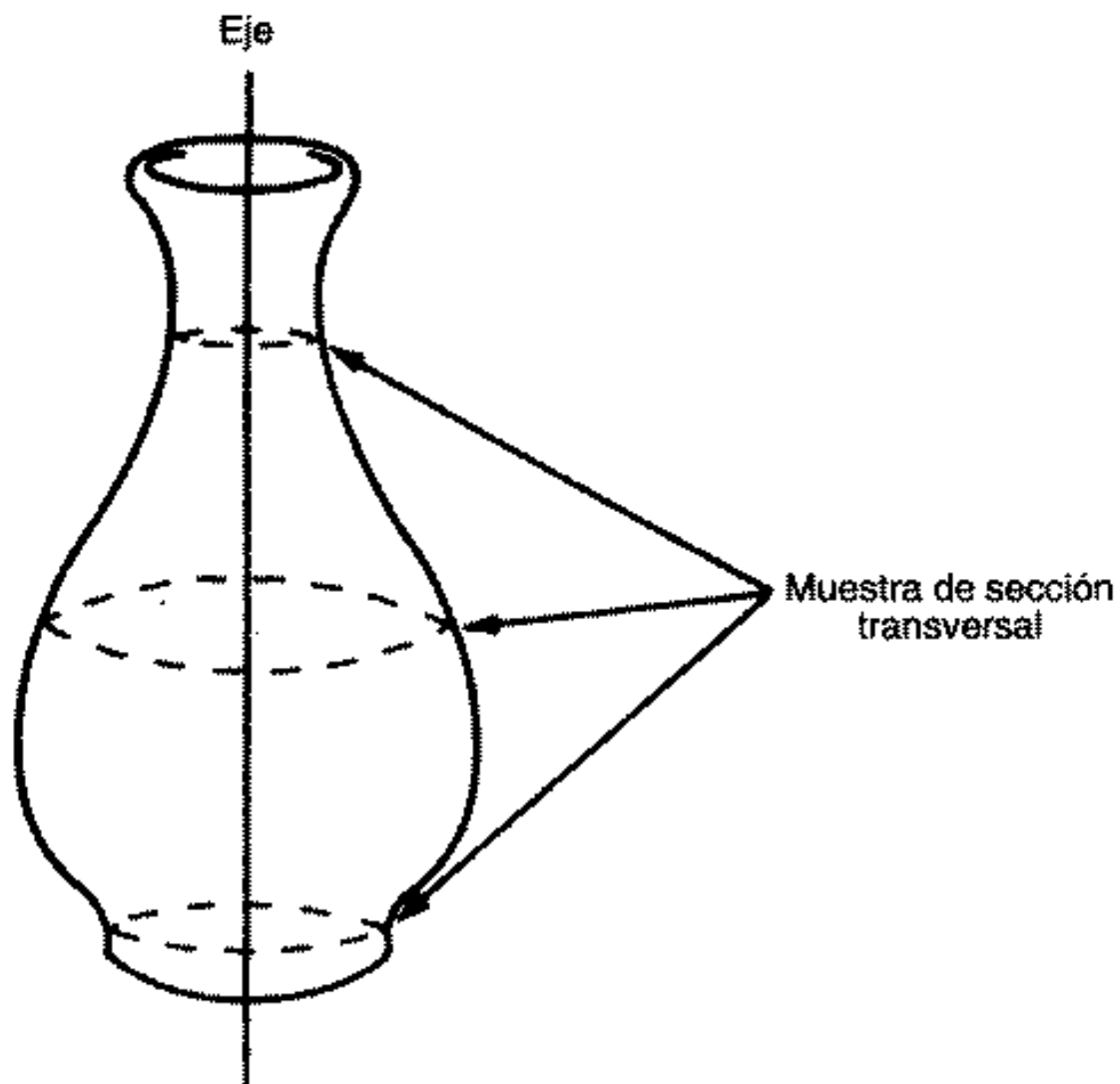


Figura A.23: Ejemplo de cono generalizado

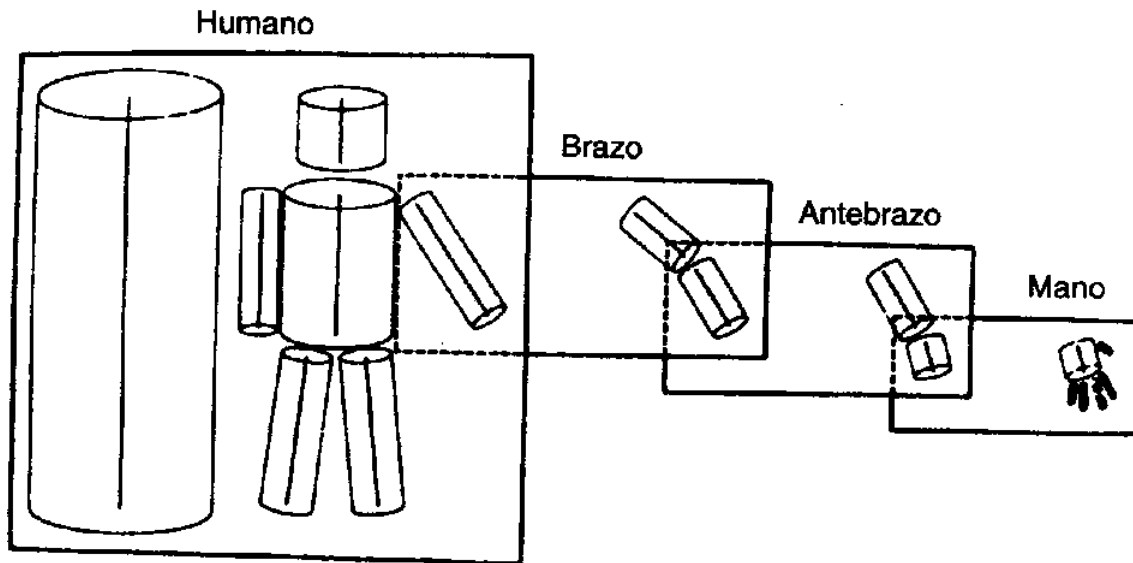


Figura A.24: Jerarquía de modelos 3D. cada encuadre muestra, a la izquierda el eje principal de la figura y a la derecha sus ejes componentes

1. Un solo eje del modelo. Este proporciona una información tosca sobre el tamaño y la orientación de la forma global descrita.
2. Las disposiciones y longitudes de los principales ejes componentes.
3. Indicaciones de los modelos 3D para las formas componentes asociadas con estos ejes componentes.

Esto conduce a una jerarquía de modelos 3D ilustrada en la figura A.24.

Se piensa que el reconocimiento se logra cuando se establece una comparación entre la descripción del modelo 3D derivada de la imagen y otra del catálogo almacenado de descripciones de modelos 3D correspondientes a objetos conocidos. Estas a su vez pueden estar organizadas jerárquicamente en cuanto a la especificidad de sus descripciones (ver figura A.25). Así una figura humana puede compararse al modelo general de un bípedo, o al modelo más específico de un humano. Un mono y un humano se distinguen por las longitudes relativas de los ejes componentes en la descripción del modelo de un bípedo.

Es obvio que necesitamos abordar la cuestión de como obtener tales descripciones de modelos 3D antes del acceso al catálogo. Necesitamos obtener los ejes, a partir de una imagen, sin saber que objeto es el que la imagen representa. Marr prueba que bajo algunas condiciones estos ejes se pueden obtener a partir de los contornos oclusivos (los

que muestran la silueta del objeto) de una imagen.

Estos ejes, obtenidos a partir de los contornos oclusivos, ahora están centrados en el observador. Dependen de la imagen, que a su vez depende del punto panorámico. Debemos transformarlos en ejes centrados en el objeto y para ello tenemos que usar el procesado en el espacio de la imagen. El procesado en el espacio de la imagen actúa sobre los ejes centrados en el observador y los convierte en coordenadas centradas en el objeto, para que las relaciones entre los diferentes ejes de la figura se especifiquen en tres y no en dos dimensiones. Puede hacerse uso de información a partir de la estereopsis, la textura y el sombreado para lograr esto, pero puede también ser necesario utilizar comparaciones preliminares con descripciones del modelo 3D almacenadas, para mejorar el análisis de la imagen. Estas ideas han tenido evolución con los años que no discutiremos aquí.

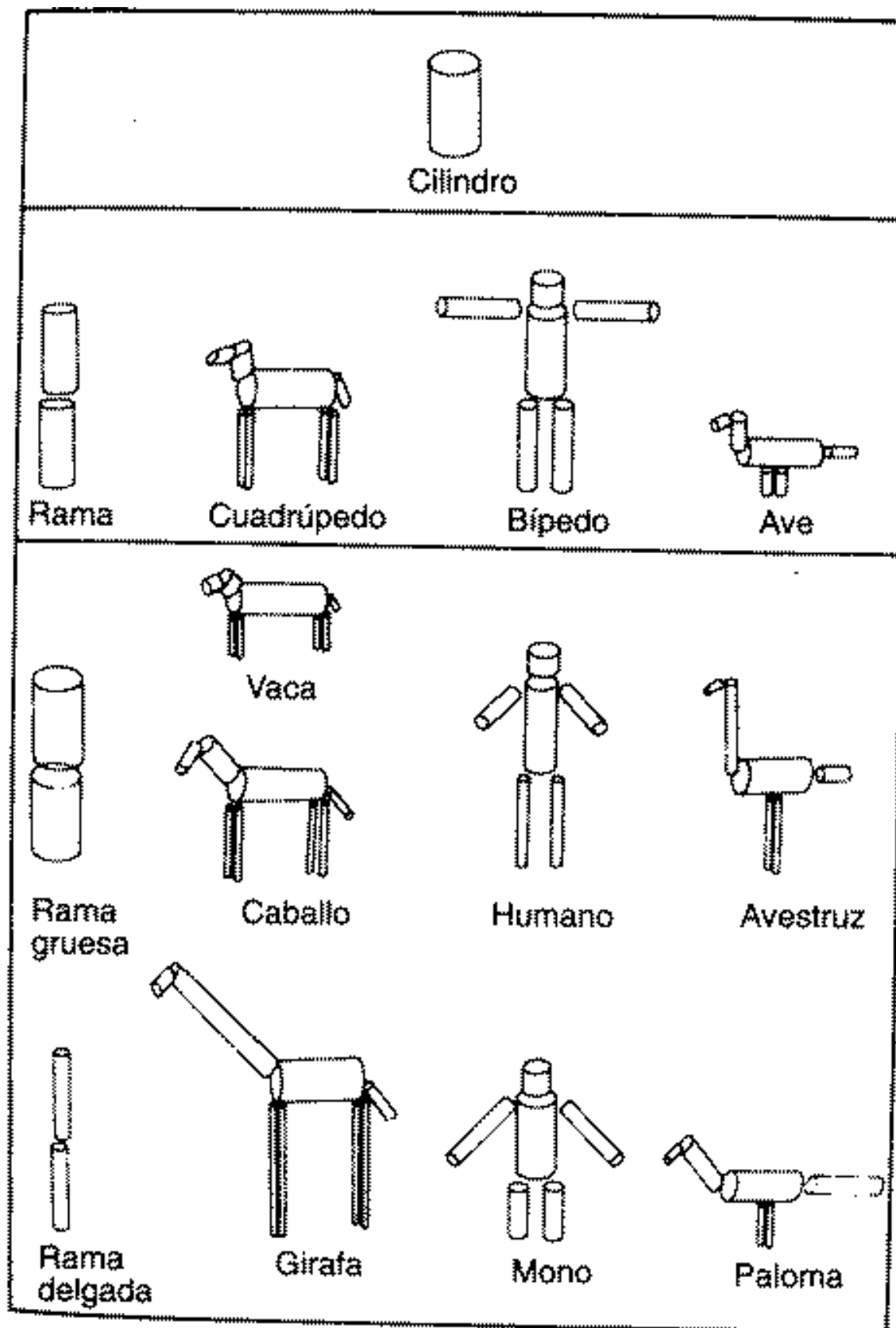


Figura A.25: Catálogo de descripciones de modelo 3D con diferentes niveles de especificidad

Apéndice B

De la Escena 3-D a la Imagen 2-D

B.1. Introducción

En este capítulo estudiaremos primero como se forman las imágenes 2-D a partir de las escenas tridimensionales. Entender la formación de imágenes es un prerequisite para entender completamente los métodos de recuperar información 3-D a partir de imágenes. Si entendemos el proceso por el cual el mundo 3-D (una escena) se proyecta en el mundo 2-D (una imagen) comprenderemos dos aspectos fundamentales de la formación de imágenes:

- ¿Qué determina donde aparece la imagen de un punto?.
- ¿Qué determina lo brillante que la imagen de una superficie será?.

La respuesta a estas preguntas requiere conocimiento sobre la proyección de imágenes, fotometría y radiometría, conceptos que discutiremos en casos simples.

B.2. Formación de Imágenes

Antes de analizar una imagen hemos de estudiar cómo se forma. Una imagen es un patrón bidimensional de luminosidad. Cómo se produce este patrón lo vamos a estudiar en dos partes. Primero veremos la correspondencia geométrica entre puntos en la escena y puntos en la imagen para posteriormente analizar qué determina la luminosidad de un punto en la imagen.

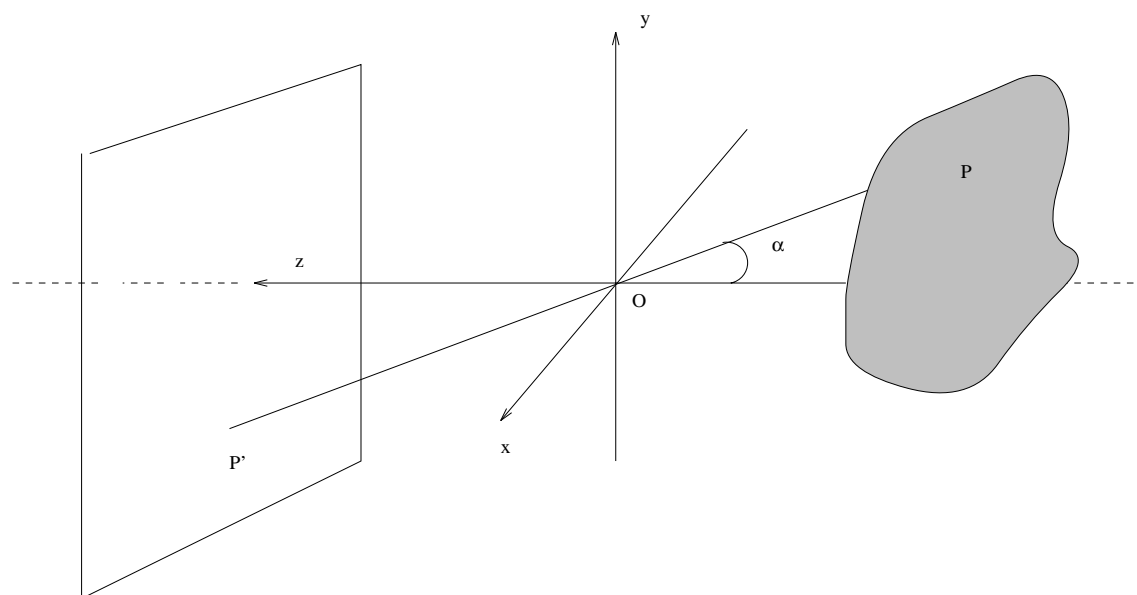


Figura B.1: De escena 3-D a la imagen 2-D

B.2.1. Perspectiva por Proyección

Consideremos un agujero diminuto ideal (ojo de aguja) a una distancia fija delante del plano de la imagen (ver figura B.1). Supongamos que idealmente sólo la luz que viene a partir del ojo de la aguja puede alcanzar el plano de la imagen. Puesto que la luz viaja en línea recta, cada punto en la imagen corresponde a una dirección particular definida por un rayo desde el punto a través del ojo de la aguja.

Definimos el eje óptico, en este caso simple, como la perpendicular, a través del ojo de la aguja, al plano de la imagen. Introducimos un sistema de coordenadas con el origen en el ojo de la aguja y el eje z alineado con el eje óptico y apuntando hacia la imagen. Es importante tener en cuenta que con esta orientación los puntos hacia los que mira la cámara tienen componente z negativa.

Queremos ahora calcular dónde aparecerá la imagen P' del punto P que está sobre un objeto enfrente de la cámara (ver figura B.1). Supondremos que no existe ningún otro objeto en el rayo que une P y el ojo de la aguja O . Sea $\mathbf{r} = (x, y, z)^t$ el vector que une P y O y $\mathbf{r}' = (x', y', f')$ el que une O y P' .

Si el rayo que conecta P y P' forma un ángulo α con el eje óptico tendremos

$$\cos(\alpha) = -\frac{\mathbf{r} \cdot \hat{\mathbf{z}}}{\|\mathbf{r}\|}$$

donde $\hat{\mathbf{z}}$ es el vector unitario en la dirección del eje óptico. Es importante recordar que los puntos delante de la cámara tienen componente z negativa. Además

$$\cos(\alpha) = \frac{f'}{\|\mathbf{r}'\|}$$

de modo que

$$\frac{\|\mathbf{r}'\|}{f'} = -\frac{\|\mathbf{r}\|}{\mathbf{r}\hat{\mathbf{z}}} \quad (\text{B.1})$$

lo que en componentes significa

$$\frac{x'}{f'} = \frac{x}{z} \quad \text{y} \quad \frac{y'}{f'} = \frac{y}{z}.$$

No discutiremos aquí el modelo de proyección ortográfica. Este tipo de proyección se modela mediante rayos paralelos al eje óptico en lugar de rayos que pasan a través del ojo de la aguja.

B.3. Introducción a la Fotometría

El tema más difícil y también más interesante de la formación de imágenes es saber que determina la luminosidad, concepto informal que se usa en lugar de luminancia e iluminancia, en un punto dado de la imagen. La luminosidad es un concepto informal que se usa para referirse al menos a dos ideas distintas, luminosidad de la escena y luminosidad de la imagen. En la imagen, la luminosidad se relaciona con el flujo de energía incidente en el plano de la imagen y puede medirse de diferentes formas. Aquí introduciremos el término *iluminancia* que sustituye al concepto informal de luminosidad de la imagen.

Comenzaremos definiendo antes lo que se conoce como **Intensidad luminosa de una fuente puntual**, ver [7] para una introducción a la fotometría.

B.3.1. Intensidad luminosa de una fuente puntual (I)

Supongamos que un emisor puntual P, ver figura B.2, emite en un ángulo sólido $d\omega$ un flujo luminoso dF . Entonces el cociente

$$I = \frac{dF}{d\omega} \quad (\text{B.2})$$

Recibe el nombre de intensidad. La unidad es la candela, cd , $1cd = 1lum/1estereorradian$. Es importante tener en cuenta que este concepto podría hacerse dependiente de la dirección.

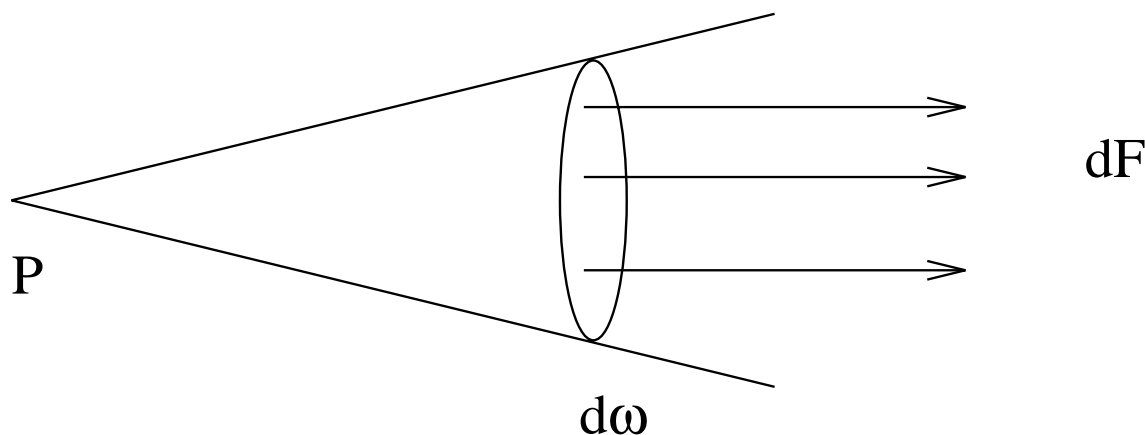


Figura B.2: intensidad

Iluminancia (E) cuando sobre un elemento de área dS llega un flujo luminoso dF , se dice que dS recibe una iluminación, a cuyo fenómeno le corresponde una magnitud llamada *iluminancia*, definida por

$$E = dF/dS \quad (\text{B.3})$$

La unidad es el lux, lx , $1lx = 1lm/1m^2$

B.3.2. Ley del cuadrado de la distancia. Relación Intensidad-Iluminancia

Si por un cono de ángulo sólido $d\omega$ se propaga un flujo dF procedente de una fuente puntual P , ver figura B.3, la intensidad correspondiente será $I = dF/d\omega$.

Si este flujo incide sobre una superficie dS cuya normal forma con el eje del cono un ángulo α , la iluminancia que recibe será

$$E = dF/dS = (Id\omega)/dS = I \cos\alpha/r^2 \quad (\text{B.4})$$

ya que teniendo en cuenta que el ángulo sólido asociado a un elemento de área viene definido por el cono que tiene como vértice el punto de radiación y cuyo eje es el segmento de línea que va desde el punto al centro del elemento de área, y que el tamaño del ángulo sólido es el área interceptada por el cono sobre una esfera de radio unidad centrada en el punto de radiación. Puede comprobarse que $d\omega = \frac{dS \cos \alpha}{r^2}$.

La ecuación anterior contiene dos leyes: primero, la iluminancia que sobre un elemento de superficie produce el flujo procedente de una fuente puntual es inversamente propor-

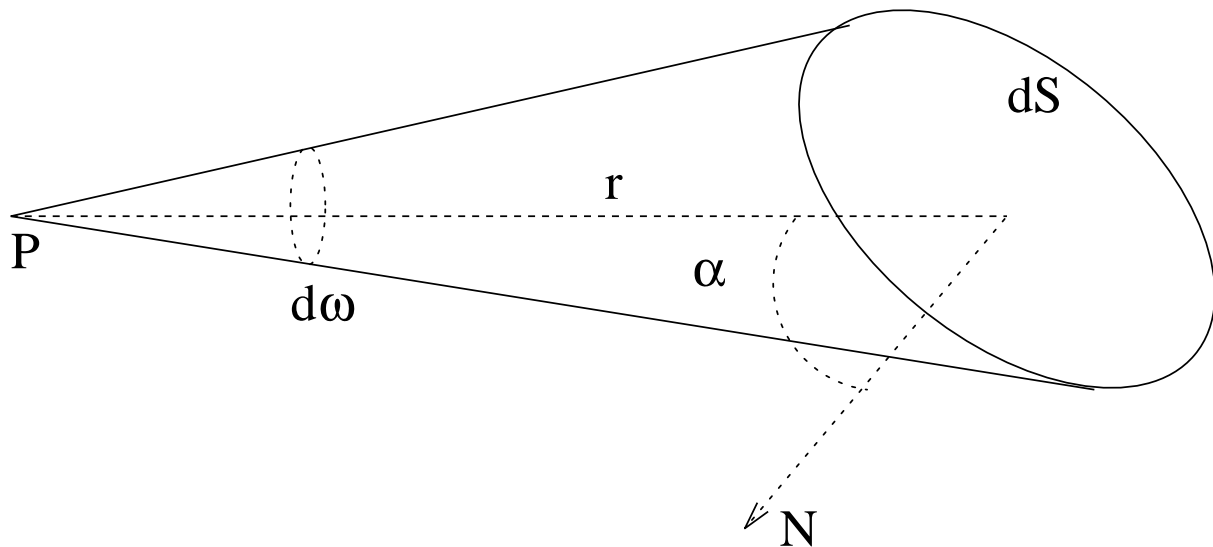


Figura B.3: Intensidad-Iluminancia

cional al cuadrado de la distancia, y segundo, la iluminancia es directamente proporcional al coseno del ángulo de incidencia.

B.3.3. Fuentes no Puntuales. Luminancia

En general el punto no tiene existencia sino como límite de una superficie o un volumen. Las fuentes luminosas tendrán una extensión superficial por pequeña que sea. Lo mismo es aplicable a los receptores. Por tanto hemos de hacer una fotometría para superficies.

Luminancia. Esta magnitud fotométrica tiene una importancia excepcional ya que es la variable que aprecia el ojo cuando observa fuentes extensas. Para definirla supongamos (ver figura B.4), un elemento de superficie dS que emite luz en todas las direcciones con uniformidad en todos sus puntos y un receptor de área dS' a distancia r de dS . Supongamos también que la normal N a dS en P forma un ángulo α con r , que dS' es normal a r , y que las dimensiones lineales de dS y dS' son pequeñas con respecto a r .

De cada punto de dS saldrá un cono elemental de luz, cuyo flujo recogerá el receptor dS' . Dadas las pequeñas dimensiones lineales de dS y dS' frente a r , estos conos pueden considerarse de igual ángulo sólido $d\omega$, y que todos transportan el mismo flujo, supuesta la emisión uniforme.

Es obvio que el flujo total que dF que dS' recibe de dS será proporcional a dS y $d\omega$

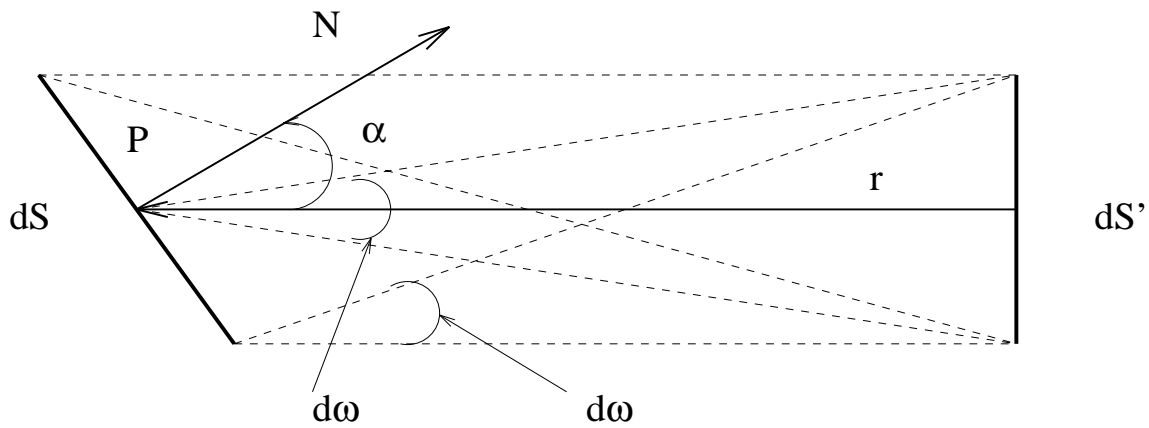


Figura B.4: Fuentes no puntuales

además, es obvio que dicho flujo dependerá en general del ángulo α a través de un factor $f(\alpha)$ a determinar en cada caso. Todo lo anterior es expresable mediante la ecuación

$$dF = L_{\alpha} dS d\omega f(\alpha) \quad (\text{B.5})$$

el coeficiente L_{α} se llama luminancia de la superficie emisora en la dirección definida por el ángulo α , y podemos escribir

$$L_{\alpha} = dF / (dS d\omega f(\alpha)) \quad (\text{B.6})$$

Cuando dS' es la pupila del ojo, tenemos una sensación subjetiva que recibe el nombre de claridad, la cual varía con la luminancia. Iguales luminancias producen la misma claridad subjetiva.

B.3.4. Emisores y Difusores perfectos. Ley de Lambert

Supongamos una superficie emisora dS limitada por un diafragma elemental D . Fijando α y r y modificando D y dS' podemos observar la proporcionalidad de dF con respecto a dS y $d\omega$. Fijando D y dS' y girando dS podemos determinar el producto $L_{\alpha} f(\alpha)$ (ver figura B.5).

Existen ciertos emisores y difusores para los que $L_{\alpha} f(\alpha)$ es $L \cos(\alpha)$. Son los llamados emisores o difusores perfectos y también lambertianos. Estas superficies presentan la misma claridad independiente del ángulo de observación. Para ellos se tiene

$$dF = L dS d\omega \cos(\alpha) \quad (\text{B.7})$$

$$L = dF / (dS d\omega \cos(\alpha)) \quad (\text{B.8})$$

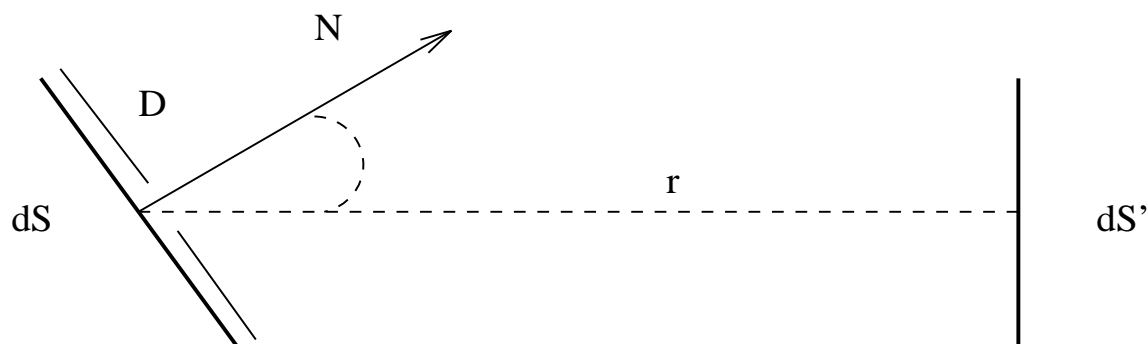


Figura B.5: Emisores y Difusores

observemos que $dF/d\omega$ tiene las dimensiones de una intensidad. Es la intensidad fuente extensa en la dirección α respecto a su normal, I_α y se tiene

$$L = I_\alpha / (dS \cos(\alpha))$$

su unidad es el nit, $1 \text{ nit} = 1 \text{ cd}/\text{m}^2$.

B.3.5. Ley de Lambert

Cuando una superficie emisora se observa desde distintos ángulos, se ve para muchos emisores y difusores que la claridad que presenta es independiente del ángulo de observación. Esto quiere decir, como ya hemos visto que estamos en presencia de superficies lambertianas. Si llamamos I_0 a la intensidad según la normal, e I_α a la que presenta en la dirección α podemos escribir

$$L = \frac{I_0}{dS} = \frac{I_\alpha}{dS \cos(\alpha)}$$

de donde tenemos la llamada ley de Lambert

$$I_\alpha = I_0 \cos(\alpha)$$

Todo emisor o difusor que cumpla esta ley se dice que emite con arreglo a la ley del coseno o ley de Lambert.

Los metales fundidos cumplen la ley del coseno con bastante aproximación. Como difusor patrón se suele utilizar el óxido de magnesio. También son buenos difusores los vidrios esmerilados, las porcelanas blancas esmeriladas y el yeso.

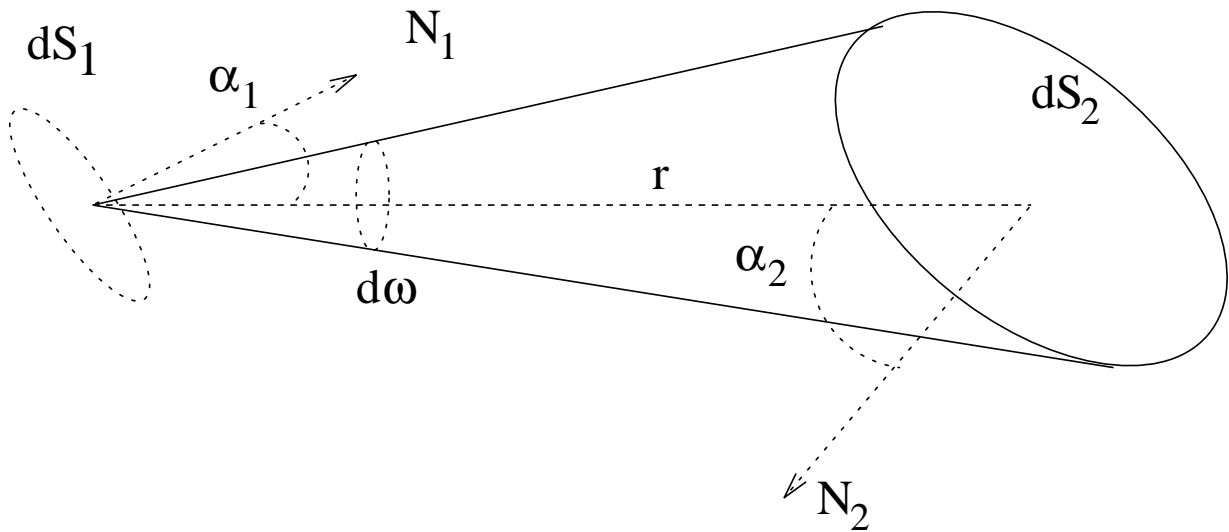


Figura B.6: Relaciones fotométricas entre superficies

En adelante trataremos con emisores difusores perfectos y emplearemos para su luminancia constante el símbolo L .

Antes de analizar las relaciones fotométricas más importantes es bueno comentar las relaciones entre las emisiones puntuales y de fuentes extensas. Puede probarse que la intensidad para fuentes extensas concuerda funcionalmente con la intensidad definida para objetos puntuales, siendo válida también la ley del cuadrado de la distancia, ver [7].

B.3.6. Relaciones Fotométricas

Del contenido de los párrafos anteriores se deducen una serie de importantes relaciones entre las distintas magnitudes fotométricas que ahora estudiaremos.

1. Relación Intensidad-Luminancia. Es obvio que se tiene

$$I_{\alpha} = LdS\cos(\alpha)$$

2. Relaciones Fotométricas entre dos elementos de superficie. Supongamos, figura B.6, dos elementos de superficie dS_1 y dS_2 , siendo r la distancia entre ambos y que la recta que los une forma con sus vectores normales unos ángulos α_1 y α_2 respectivamente.

Como ya hemos visto, el flujo que de dS_1 recibe dS_2 , supuesto un emisor perfecto, es

$$dF = I_{\alpha_1} d\omega = I_0 \cos(\alpha_1) d\omega \quad (\text{B.9})$$

$$= L_1 \cos(\alpha_1) dS_1 d\omega \quad (\text{B.10})$$

$$= \frac{L_1 \cos(\alpha_1) dS_1 \cos(\alpha_2) dS_2}{r^2} \quad (\text{B.11})$$

siendo por tanto la iluminancia en dS_2

$$E = dF/dS_2 = \frac{L_1 \cos(\alpha_1) dS_1 \cos(\alpha_2)}{r^2}$$

3. Relación Flujo-Luminancia. Supongamos un elemento de emisor perfecto dS , ver figura B.7, que emite un flujo total F en el semiespacio superior y que presenta una luminancia L , observemos que estamos hablando de emitir y observar. Tomemos como elemento de superficie $d\sigma$, la zona de amplitud $d\alpha$ cuya área será $d\sigma = 2\pi r^2 \sin(\alpha) d\alpha$, el ángulo sólido que subtiende $d\sigma$ desde el centro de la esfera será $d\omega = d\sigma/r^2 = 2\pi \sin(\alpha) d\alpha$, éste es el ángulo sólido del anillo que determina $d\alpha$.

Si llamamos I_α a la intensidad en la dirección α , tendremos para el flujo que dS emite en $d\omega$

$$dF = I_\alpha d\omega = L dS \cos(\alpha) d\omega = 2\pi L dS \sin(\alpha) \cos(\alpha) d\alpha$$

integrando para toda la semiesfera

$$F = 2\pi L dS \int_0^{2\pi} \sin(\alpha) \cos(\alpha) d\alpha = \pi L dS \quad (\text{B.12})$$

4. Llegamos ahora a la relación más importante, la que liga el flujo recibido con el flujo emitido. Relación Iluminancia-Luminancia.

Supongamos que un elemento difusor perfecto dS recibe un flujo F y lo reemite totalmente con arreglo a la ley del coseno. Se comportará entonces como un difusor perfecto y cumplirá la ecuación B.12.

Por otra parte, la iluminación recibida será $E = F/dS$, obteniéndose entonces la siguiente relación entre la iluminancia y la luminancia

$$E = \pi L$$

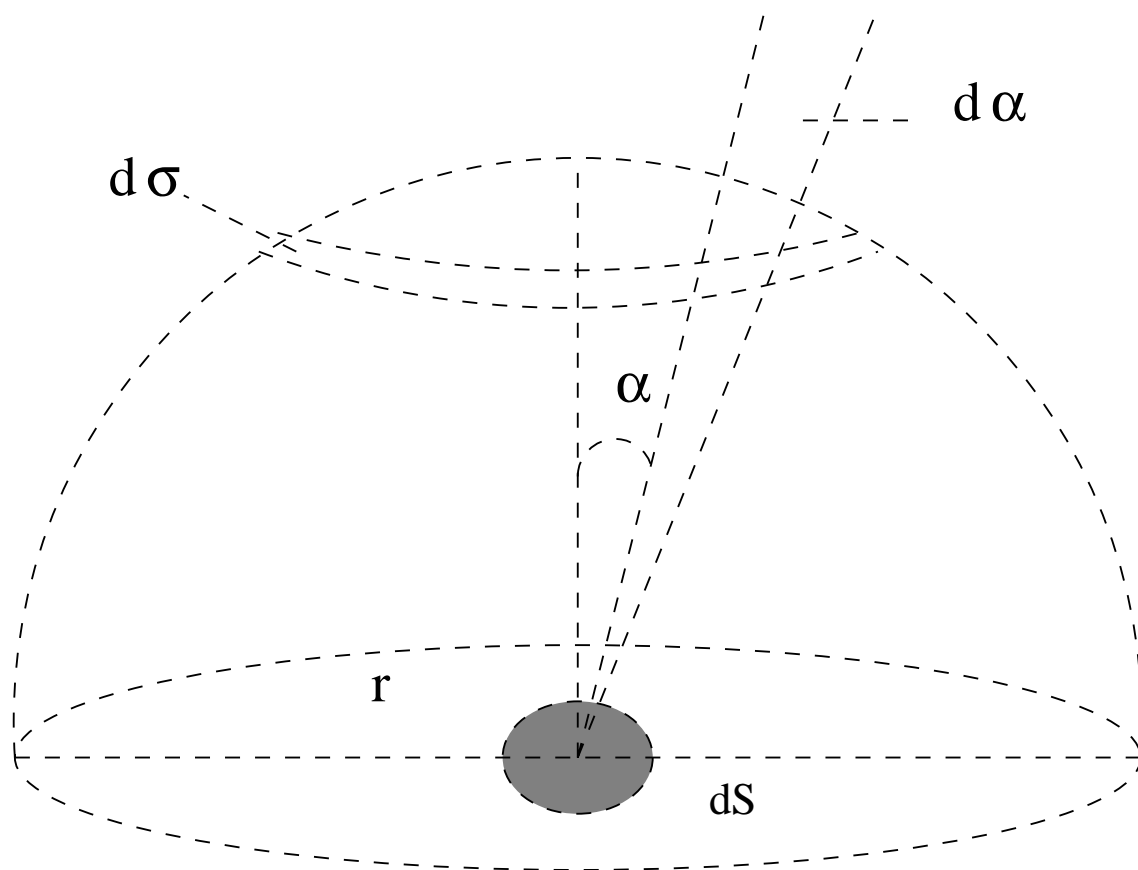


Figura B.7: Relación flujo-lumimancia

si el difusor tiene un factor de reflectancia ρ la ecuación anterior se convierte en

$$\rho E = \pi L$$

Por último es importante comentar aquí y brevemente el concepto de función de distribución de reflectancia bidimensional. Esta función se define como la fracción de luz incidente que se emite en una dirección cuando la superficie es iluminada desde otra (ver [18]). Esta función la base de muchas técnicas para la recuperación de información tridimensional a partir de imágenes.

Apéndice C

Preliminares Matemáticos

C.1. Resultados de Teoría de Matrices

C.1.1. Vectores y Matrices

Con frecuencia, las sucesiones unidimensionales y bidimensionales se representan mediante vectores y matrices, respectivamente. Un vector columna, u , que contiene N elementos se representa como

$$u = \{u(n)\} = \begin{bmatrix} u(1) \\ u(2) \\ \vdots \\ u(N) \end{bmatrix}$$

El n -ésimo elemento del vector u se nota $u(n)$, u_n , o $[u]_n$. Un vector columna de tamaño N también se denomina vector $N \times 1$. De forma análoga, un vector fila de tamaño N se denomina vector $1 \times N$.

Una matriz A de tamaño $M \times N$ tiene M filas y N columnas y se define como

$$A = \{a(m, n)\} = \begin{bmatrix} a(1, 1) & a(1, 2) & \dots & a(1, N) \\ a(2, 1) & a(2, 2) & \dots & a(2, N) \\ \vdots & \vdots & & \vdots \\ a(M, 1) & a(M, 2) & \dots & a(M, N) \end{bmatrix}$$

El elemento de la m -ésima fila y la n -ésima columna de la matriz A se nota $[A]_{m,n} = a(m, n) = a_{m,n}$. La n -ésima columna de A se nota a_n , cuyo n -ésimo elemento se nota $a_n(m) = a(m, n)$.

Algunas definiciones básicas de teoría de matrices son las siguientes

Concepto	Definición	Observaciones
Matriz	$A = \{a(m, n)\}$	$m =$ índice de filas, $n =$ índice de columnas
Traspuesta	$A^T = \{a(n, m)\}$	
Conjugada compleja	$A^* = \{a^*(m, n)\}$	
Traspuesta conjugada	$A^{*T} = \{a^*(n, m)\}$	
Matriz identidad	$I = \{\delta(m - n)\}$	
Matriz nula	$0 = \{0\}$	
Suma	$A + B = \{a(m, n) + b(m, n)\}$	A y B son $M \times N$
Multiplicación escalar	$\alpha A = \{\alpha a(m, n)\}$	
Multiplicación matricial	$c(m, n) = \sum_{k=1}^K a(m, k)b(k, n)$	A es $M \times K$, B es $K \times N$, C es $M \times N$. $AB \neq BA$
Matrices que conmutan	$AB = BA$	En general, no es cierto
Producto interior	$\langle x, y \rangle = x^{*T}y = \sum_n x^*(n)y(n)$	Escalar. Si $\langle x, y \rangle = 0$, x e y son ortogonales
Producto exterior	$xy^T = \{x(m)y(n)\}$	x es $M \times 1$, y es $N \times 1$, el producto exterior es $M \times N$
Matriz simétrica	$A = A^T$	
Matriz de Hermite	$A = A^{*T}$	Una matriz real simétrica es una matriz de Hermite. Sus autovalores son reales
Determinante	$ A $	Sólo para matrices cuadradas
Rango, $[A]$	Número de filas o columnas linealmente independientes	
Inversa, A^{-1}	$A^{-1}A = AA^{-1} = I$	Sólo para matrices cuadradas
Matriz singular	A^{-1} no existe	$ A = 0$
Traza	$Tr[A] = \sum_n a(n, n)$	
Autovalores, λ_k	Las raíces de $ A - \lambda_k I = 0$	
Autovectores, Φ_k	Las soluciones de $A\Phi_k = \lambda_k \Phi_k, \Phi \neq 0$	

C.1.2. Ordenación por Filas y por Columnas

A veces es necesario expresar una matriz en forma de vector, por ejemplo, cuando se almacena una imagen en un disco o una cinta. Sea $x = \mathcal{O}\{x(m, n)\}$ una ordenación de los elementos de la matriz $\{x(m, n)\}$ en el vector x .

Una organización muy empleada es la llamada **ordenación lexicográfica**. La **ordenación por filas** de una matriz, $X_{M \times N}$, se define como

$$x^T = [x(1, 1), \dots, x(1, N), x(2, 1), \dots, x(2, N), \dots, x(M, 1), \dots, x(M, N)]^T = \mathcal{O}_r\{x(m, n)\} \quad (\text{C.1})$$

Por lo tanto, x^T es el vector fila obtenido colocando cada fila de X a la derecha de la anterior.

Otra organización útil es la **ordenación por columnas**, que da el siguiente vector ordenado por columnas

$$x^T = [x(1, 1), \dots, x(M, 1), x(1, 2), \dots, x(M, 2), \dots, x(1, N), \dots, x(M, N)]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \mathcal{O}_c\{x(m, n)\} \quad (\text{C.2})$$

donde x_n es la n -ésima columna de la matriz X .

C.1.3. Reglas de Trasposición y Conjugación

- $A^{*T} = [A^T]^*$
- $[AB]^T = B^T A^T$
- $[A^{-1}]^T = [A^T]^{-1}$
- $[AB]^* = A^* B^*$

C.1.4. Matrices Circulantes y de Toeplitz

Una **matriz de Toeplitz**, T , es una matriz que tiene elementos constantes a lo largo de la diagonal principal y las subdiagonales. Esto significa que los elementos $t(m, n)$ dependen solamente de la diferencia $m - n$, por lo que notaremos $t(m, n) = t_{m-n}$. Por lo tanto una matriz de Toeplitz $N \times N$ es de la forma

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-N+1} \\ t_1 & t_0 & t_{-1} & \cdots & t_{-N+2} \\ t_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & t_{-1} \\ t_{N-1} & \cdots & t_2 & t_1 & t_0 \end{bmatrix}$$

y está completamente definida mediante los $2N - 1$ elementos $\{t_k, -N + 1 \leq k \leq N - 1\}$. Las matrices de Toeplitz describen las transformaciones entrada-salida de los sistemas unidimensionales lineales espacialmente invariantes.

Una matriz, C , es **circulante** si cada una de sus filas (o columnas) se obtiene mediante un desplazamiento circular de la fila (o columna) anterior

$$T = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & c_1 & \cdots & c_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & & \ddots & \ddots & c_1 \\ c_1 & c_2 & \cdots & c_{N-1} & c_0 \end{bmatrix}$$

Podemos comprobar que C también es Toeplitz y que $c(m, n) = c((m - n) \bmod N)$.

Las matrices circulantes describen el comportamiento entrada-salida de los sistemas unidimensionales lineales periódicos.

C.1.5. Matrices Ortogonales y Unitarias

Una **matriz ortogonal** es aquella cuya inversa es igual a su traspuesta, es decir, si $A^{-1} = A^T$, o $A^T A = A A^T = I$.

Se dice que una matriz es **unitaria** si su inversa es igual a su traspuesta conjugada, es decir, si $A^{-1} = A^{*T}$, o $A A^{*T} = A^{*T} A = I$.

Una matriz real ortogonal también es unitaria, pero una matriz unitaria no tiene por qué ser ortogonal. Las definiciones anteriores implican que las columnas (o filas) de una

matriz unitaria $N \times N$ son ortogonales y forman un conjunto completo de vectores base de un espacio vectorial N -dimensional.

C.1.6. Matrices Definidas Positivas y Formas Cuadráticas

Se dice que una matriz de Hermite, A , de tamaño $N \times N$ es **definida positiva** o **semidefinida positiva** si la forma cuadrática

$$Q = x^{*T} A x, \quad \forall x \neq 0$$

es positiva (> 0) o no negativa (≥ 0), respectivamente. De forma análoga, A es **definida negativa** o **semidefinida negativa** si $Q < 0$ o $Q \leq 0$. Una matriz que no satisface ninguna de las anteriores condiciones es **indefinida**.

Si A es una matriz simétrica definida positiva (no negativa), todos sus autovalores $\{\lambda_k\}$ son positivos (no negativos) y el determinante de A satisface la desigualdad

$$|A| = \prod_{k=1}^N \lambda_k \leq \prod_{k=1}^N a(k, k) \quad (\text{C.3})$$

C.1.7. Formas Diagonales

Para cualquier matriz de Hermite, R , existe una matriz unitaria Φ tal que

$$\Phi^{*T} R \Phi = \Lambda \quad (\text{C.4})$$

donde Λ es una matriz diagonal que contiene los autovalores de R . Una forma alternativa de (C.4) es

$$R \Phi = \Phi \Lambda \quad (\text{C.5})$$

que es el conjunto de ecuaciones de autovalores

$$R \Phi_k = \lambda_k \Phi_k \quad k = 1, \dots, N \quad (\text{C.6})$$

donde $\{\lambda_k\}$ y $\{\Phi_k\}$ son los autovalores y autovectores de R . Para matrices de Hermite, los autovectores correspondientes a autovalores distintos son ortogonales. Para autovalores repetidos, sus autovectores forman un subespacio que puede ser ortogonalizado para obtener un conjunto completo de autovectores ortogonales. La normalización de estos autovectores da un conjunto ortonormal, es decir, la matriz unitaria Φ , cuyas columnas son estos autovectores. La matriz Φ se denomina **automatriz**.

C.2. Matrices por Bloques y Matrices de Kronecker

En procesamiento de imágenes, el análisis de muchos problemas se puede simplificar sustancialmente trabajando con matrices por bloques y los llamados **productos de Kronecker**. Por ejemplo, la convolución bidimensional puede expresarse a través de operaciones simples de matrices por bloques.

C.2.1. Matrices por Bloques

Cualquier matriz, A , cuyos elementos son a su vez matrices se denomina **matriz por bloques**; por ejemplo

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & & \vdots \\ A_{m,1} & A_{m,2} & \cdots & A_{m,n} \end{bmatrix}$$

es una matriz por bloques, donde $\{A_{i,j}\}$ son matrices $p \times q$. Se dice que la matriz A es una matriz por bloques $m \times n$ de dimensión básica $p \times q$. Si $A_{i,j}$ son matrices cuadradas ($p \times p$), se dice que A es una matriz por bloques $m \times n$ de dimensión básica p .

Si la estructura de los bloques es de Toeplitz ($A_{i,j} = A_{i-j}$), o circulante ($A_{i,j} = A_{((i-j) \bmod n)}, m = n$) entonces A se denomina **Toeplitz por bloques** o **circulante por bloques**, respectivamente. Además, si cada bloque a su vez es Toeplitz (o circulante), entonces A se denomina **doblemente Toeplitz por bloques** (o **doblemente circulante por bloques**). Si $\{A_{i,j}\}$ son Toeplitz (o circulantes) pero ($A_{i,j} \neq A_{i-j}$) entonces A se denomina **matriz de bloques Toeplitz** (o **de bloques circulantes**). Una matriz doblemente Toeplitz (o circulante) por bloques no tiene por qué ser totalmente Toeplitz (o circulante).

C.2.2. Productos de Kronecker

Si A y B son matrices $M_1 \times M_2$ y $N_1 \times N_2$, su producto de Kronecker se define como

$$A \otimes B = \{a(m,n)B\} = \begin{bmatrix} a(1,1)B & \cdots & a(1,M_2)B \\ \vdots & & \vdots \\ a(M_1,1)B & \cdots & a(M_1,M_2)B \end{bmatrix} \quad (\text{C.7})$$

Esta es una matriz por bloques $M_1 \times M_2$ de dimensión básica $N_1 \times N_2$. Resulta evidente que $A \otimes B \neq B \otimes A$. Los productos de Kronecker son útiles para generar matrices de mayor orden a partir de otras de orden menor. Algunas propiedades de los productos de Kronecker son las siguientes

1. $(A + B) \otimes C = A \otimes C + B \otimes C$
2. $(A \otimes B) \otimes C = A \otimes (B \otimes C)$
3. $\alpha(A \otimes B) = (\alpha A) \otimes B = A \otimes (\alpha B)$, donde α es un escalar.
4. $(A \otimes B)^T = A^T \otimes B^T$
5. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$
6. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$
7. $A \otimes B = (A \otimes I)(I \otimes B)$
8. $\prod_{k=1}^l (A_k \otimes B_k) = \left(\prod_{k=1}^l A_k \right) \otimes \left(\prod_{k=1}^l B_k \right)$, donde A_k y B_k son matrices cuadradas.
9. $\det(A \otimes B) = (\det A)^m (\det B)^n$, donde A es $m \times m$ y B es $n \times n$
10. $Tr(A \otimes B) = [Tr(A)][Tr(B)]$
11. Si $r(A)$ denota el rango de una matriz A , entonces $r(A \otimes B) = r(A)r(B)$.
12. Si A y B son unitarias, entonces $A \otimes B$ también es unitaria.
13. Si $C = A \otimes B$, $Ct_k = \gamma_k t_k$, $Ax_i = \lambda_i x_i$, $By_j = \mu_j y_j$,
entonces $t_k = x_i \otimes y_j$, $\gamma_k = \lambda_i \mu_j$, $1 \leq i \leq m$, $1 \leq j \leq n$, $1 \leq k \leq mn$.

C.2.3. Operaciones Separables

Consideremos la transformación (en una imagen U de tamaño $N \times M$)

$$V = AUB^T \Leftrightarrow v(k, l) = \sum_m \sum_n a(k, m)u(m, n)b(l, n) \quad (\text{C.8})$$

Esta expresión define una clase de operaciones separables, donde A opera sobre las columnas de U y B opera sobre las filas del resultado. Si v_k y u_m denotan los vectores fila k -ésimo y m -ésimo de V y U , respectivamente, la serie anterior se puede expresar como

$$v_k^T = \sum_m a(k, m)[Bu_m^T] = \sum_m [A \otimes B]_{k,m} u_m^T \quad (\text{C.9})$$

donde $[A \otimes B]_{k,m}$ es el (k, m) -ésimo bloque de $A \otimes B$. Por lo tanto, si U y V están ordenadas por filas en los vectores u y v , respectivamente, entonces

$$V = AUB^T \implies v = (A \otimes B)u \quad (\text{C.10})$$

Así pues, esta transformación separable resulta en un producto de Kronecker que opera sobre un vector.

C.3. Señales Aleatorias

C.3.1. Definiciones

Una **señal discreta aleatoria compleja** o **proceso aleatorio discreto** es una sucesión de variables aleatorias $u(n)$. Para sucesiones aleatorias complejas se definen

$$\begin{aligned} \text{Media} &= \mu_u(n) = \mu(n) = E[u(n)] \\ \text{Varianza} &= \sigma_u^2 = E[|u(n) - \mu(n)|^2] \\ \text{Covarianza} &= \text{Cov}[u(n), u(n')] = r_u(n, n') = E\{[u(n) - \mu(n)][u^*(n') - \mu^*(n')]\} \\ \text{Covarianza cruzada} &= \text{Cov}[u(n), v(n')] = r_{uv}(n, n') = E\{[u(n) - \mu_u(n)][v^*(n') - \mu_v^*(n')]\} \\ \text{Autocorrelación} &= a_{uu}(n, n') = a(n, n') = E[u(n)u^*(n')] = r(n, n') + \mu(n)\mu^*(n') \\ \text{Correlación cruzada} &= a_{uv}(n, n') = E[u(n)v^*(n')] = r_{uv}(n, n') + \mu_u(n)\mu_v^*(n') \end{aligned}$$

El operador E representa la esperanza matemática. Para un vector u de tamaño $N \times 1$, su media, covarianza y otras propiedades se definen de la siguiente forma

$$\begin{aligned} E[u] &= \mu = \{\mu(n)\} \text{ es un vector } N \times 1 \\ \text{Cov}[u] &= E(u - \mu)(u^* - \mu^*)^T = R_u = \{r(n, n')\} \text{ es una matriz } N \times N \\ \text{Cov}[u, v] &= E(u - \mu_u)(v^* - \mu_v^*) = R_{uv} = \{r_{uv}(n, n')\} \text{ es una matriz } N \times N \end{aligned}$$

μ y R representan el vector de medias y la matriz de covarianzas del vector u .

C.3.2. Distribución Normal o Gaussiana

La función de densidad de probabilidad de una variable aleatoria, u , se nota $p_u(u)$. Para una **variable aleatoria Gaussiana**

$$p_u(u) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{|u - \mu|^2}{2\sigma^2}\right\} \quad (\text{C.11})$$

donde μ y σ^2 son su media y varianza, y u representa el valor que toma la variable aleatoria. Para $\mu = 0$ y $\sigma^2 = 1$, se denomina **distribución normal estándar**.

C.3.3. Procesos Aleatorios Gaussianos

Una sucesión se denomina **proceso aleatorio gaussiano** (o **normal**) si la densidad de probabilidad asociada a cualquier subsucesión finita es una distribución gaussiana. Por ejemplo, para una sucesión gaussiana $\{u(n), 1 \leq n \leq N\}$, la densidad asociada sería

$$p_u(u) = p_u(u_1, u_2, \dots, u_N) = [(2\pi)^{N/2} |R|^{1/2}]^{-1} \exp\{-1/2(u - \mu)^* R^{-1} (u - \mu)\} \quad (\text{C.12})$$

donde R es la matriz de covarianza de u y se supone que es no singular.

C.3.4. Procesos Estacionarios

Se dice que una sucesión aleatoria, $u(n)$, es **estrictamente estacionaria** si la densidad asociada a cualquier subsucesión $\{u(l), 1 \leq l \leq k\}$ es la misma que la de la sucesión desplazada $\{u(l+m), 1 \leq l \leq k\}$, para cualquier entero m y cualquier longitud k . Se dice que la sucesión $u(n)$ es **estacionaria en un sentido amplio** si

$$E[u(n)] = \mu = \text{constante} \quad (\text{C.13})$$

$$E[u(n)u^*(n')] = r(n - n') \quad (\text{C.14})$$

Esto implica que $r(n, n') = r(n - n')$, lo que significa que la matriz de covarianza de $u(n)$ es Toeplitz.

La función de covarianza de un proceso estacionario, $u(n)$, verifica

$$r(n) = \text{Cov}[u(n), u(0)] = \text{Cov}[u(n' + n), u(n')], \quad \forall n', \forall n \quad (\text{C.15})$$

A partir de su definición puede comprobarse que las matrices de covarianza y de autocorrelación, $r(n, n')$ y $a(n, n')$, son de Hermite y definidas no negativas

$$r(n, n') = r^*(n', n), \quad \forall n, n' \quad (\text{C.16})$$

$$\sum_n \sum_{n'} x(n)r(n, n')x^*(n') \geq 0, \quad x(n) \neq 0, \quad \forall n \quad (\text{C.17})$$

C.3.5. Procesos de Markov

Se dice que una sucesión aleatoria, $u(n)$, es **una sucesión de Markov de orden p** si

$$\text{Prob}[u(n)|u(n-1), u(n-2), \dots] = \text{Prob}[u(n)|u(n-1), u(n-p)], \quad \forall n \quad (\text{C.18})$$

Una interpretación de una sucesión de Markov de orden p es que si el "presente", $\{u(j), n-p \leq j \leq n-1\}$, es conocido, entonces el "pasado", $\{u(j), j \leq n-p\}$ y el "futuro", $\{u(j), j \geq n\}$, son independientes. Por tanto la esperanza condicional satisface

$$E[u(n)|u(n-1), u(n-2), \dots] = E[u(n)|u(n-1), \dots, u(n-p)], \quad \forall n \quad (\text{C.19})$$

C.3.6. Ortogonalidad e Independencia

Se dice que dos variables aleatorias, x e y , son **independientes** sii su función de probabilidad conjunta es el producto de las probabilidades marginales

$$p_{x,y}(x, y) = p_x(x)p_y(y) \quad (\text{C.20})$$

Se dice que dos sucesiones aleatorias, $x(n)$ e $y(n)$, son **independientes** sii para todo n y n' las variables aleatorias $x(n)$ e $y(n')$ son independientes.

Se dice que las variables aleatorias x e y son **ortogonales** si

$$E[xy^*] = 0 \quad (\text{C.21})$$

y que están **incorreladas** si

$$E[xy^*] = (E[x])(E[y^*]) \quad (\text{C.22})$$

o

$$E[(x - \mu_x)(y - \mu_y)^*] = 0 \quad (\text{C.23})$$

Por lo tanto las variables incorreladas de media 0 también son ortogonales. Las variables aleatorias gaussianas que están incorreladas también son independientes.

C.4. Campos Aleatorios Discretos

Cuando cada elemento de una sucesión bidimensional es una variable aleatoria, se le denomina **campo aleatorio discreto**. Un campo aleatorio discreto puede representar, por ejemplo, un conjunto de imágenes.

La funciones de media y covarianza de un campo aleatorio complejo se definen como

$$E[u(m, n)] = \mu(m, n) \quad (\text{C.24})$$

$$\begin{aligned} \text{Cov}[u(m, n), u(m', n')] &= r_u(m, n; m', n') = \\ &= E[(u(m, n) - \mu(m, n))(u^*(m', n') - \mu^*(m', n'))] \end{aligned} \quad (\text{C.25})$$

Se dice que un campo aleatorio es **espacialmente invariante**, **homogéneo** o **estacionario** si

$$\mu(m, n) = \mu = \text{constante} \quad (\text{C.26})$$

$$r_u(m, n; m', n') = r_u(m - m', n - n') \quad (\text{C.27})$$

En este caso, la función de covarianza verifica

$$r(m, n) = \text{Cov}[u(m, n), u(0, 0)] = \text{Cov}[u(m + m', n + n'), u(m', n')] \quad \forall(m', n') \quad (\text{C.28})$$

Un campo aleatorio $x(m, n)$ se llamará **campo de ruido blanco** cuando dos elementos cualesquiera, $x(m, n)$ y $x(m', n')$, estén mutuamente incorrelados. La función de covarianza será entonces de la forma

$$r_x(m, n; m', n') = \sigma_x^2(m, n)\delta(m - m', n - n') \quad (\text{C.29})$$

Un campo aleatorio se llama **gaussiano** si todo segmento definido sobre un grid finito arbitrario es gaussiano. Esto significa que todo segmento finito de $u(m, n)$, cuando se represente en un vector será un proceso aleatorio gaussiano.

La covarianza y la autocorrelación de campos bidimensionales tienen propiedades de simetría y no negatividad similares a las de los procesos aleatorios unidimensionales

$$r(m, n; m', n') = r^*(m', n'; m, n) \quad (\text{C.30})$$

$$\sum_m \sum_n \sum_{m'} \sum_{n'} x(m, n)r(m, n; m', n')x^*(m', n') \geq 0, \quad x(m, n) \neq 0, \quad \forall(m, n) \quad (\text{C.31})$$

Funciones de covarianza separables e isotrópicas

Se dice que la función de covarianza de un campo aleatorio es **separable** cuando puede expresarse como un producto de funciones de covarianza de sucesiones unidimensionales

$$\text{Caso no estacionario: } r(m, n; m', n') = r_1(m, m')r_2(n, n') \quad (\text{C.32})$$

$$\text{Caso estacionario: } r(m, n) = r_1(m)r_2(n) \quad (\text{C.33})$$

Una función de covarianza separable estacionaria muy utilizada en procesamiento de imágenes es

$$r(m, n) = \sigma^2 \rho_1^{|m|} \rho_2^{|n|}, \quad |\rho_1| \leq 1, \quad |\rho_2| \leq 1 \quad (\text{C.34})$$

σ^2 representa la varianza del campo aleatorio y $\rho_1 = r(1, 0)/\sigma^2$, $\rho_2 = r(0, 1)/\sigma^2$ son la correlación en las direcciones m y n .

Otra función de covarianza considerada con frecuencia como más realista para muchas imágenes es la función exponencial no separable

$$r(m, n) = \sigma^2 \exp \left[-\sqrt{(\alpha_1 m^2 + \alpha_2 n^2)} \right] \quad (\text{C.35})$$

Cuando $\alpha_1 = \alpha_2 = \alpha$, la covarianza es función de la distancia euclídea $d = \sqrt{m^2 + n^2}$, es decir,

$$r(m, n) = \sigma^2 \rho^d \quad (\text{C.36})$$

donde $\rho = \exp(-|\alpha|)$. Esta función también se llama **isotrópica** o **circularmente simétrica**.

C.5. La Función de Densidad Espectral

Sea $u(m, n)$ una sucesión aleatoria estacionaria. Su **Función Generadora de Covarianza** (FGC) se define como la Z-transformada de su función de covarianza, $r_u(n)$

$$\text{FGC}\{u(m, n)\} = S_u(z_1, z_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} r_u(m, n) z_1^{-m} z_2^{-n} \quad (\text{C.37})$$

La **Función de Densidad Espectral** (FDS) se define como la transformada de Fourier de $r_u(n)$, que es la Función Generadora de Covarianza evaluada en $z = \exp(j2\pi\omega)$

$$\begin{aligned} \text{FDS}\{u(m, n)\} &= S_u(\omega_1, \omega_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} r_u(m, n) \exp[-j2\pi(\omega_1 m + \omega_2 n)] = \\ &= S_u(z_1, z_2) \Big|_{z_1=e^{j2\pi\omega_1}, z_2=e^{j2\pi\omega_2}} \end{aligned} \quad (\text{C.38})$$

Por tanto, la covarianza se puede expresar como la transformada de Fourier inversa de la Función de Densidad Espectral, es decir

$$r_u(m, n) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} S_u(\omega_1, \omega_2) \exp[j2\pi(\omega_1 m + \omega_2 n)] \quad (\text{C.39})$$

Por tanto

$$\sigma_u^2 = E[|u(m, n) - \mu|^2] = r_u(0, 0) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} S_u(\omega_1, \omega_2) \quad (\text{C.40})$$

es decir, el volumen bajo $S_u(\omega_1, \omega_2)$ es igual a la potencia media del campo aleatorio $u(m, n)$. Físicamente $S_u(\omega_1, \omega_2)$ representa la densidad de potencia en la imagen en frecuencias espaciales (ω_1, ω_2) . Por eso la Función de Densidad Espectral también se conoce como Función de Densidad del Espectro de Potencia o, simplemente, Espectro de Potencia del campo aleatorio.

Propiedades de la Función de Densidad Espectral

1. Teniendo en cuenta que la función de covarianza es simétrica conjugada se deduce que la Función de Densidad Espectral es real

$$S(\omega_1, \omega_2) = S^*(\omega_1, \omega_2)$$

Para campos aleatorios reales la Función de Densidad Espectral también es par.

2. La Función de Densidad Espectral es no negativa

$$S(\omega_1, \omega_2) \geq 0 \quad \forall \omega_1, \omega_2$$

Esta propiedad debe cumplirse, ya que la potencia no puede ser negativa.

Para un sistema espacialmente invariante cuya respuesta en frecuencia sea $H(\omega_1, \omega_2)$ y cuya entrada sea un campo aleatorio $\epsilon(m, n)$, la Función de Densidad Espectral de la salida, $u(m, n)$ viene dada por

$$S_u(\omega_1, \omega_2) = |H(\omega_1, \omega_2)|^2 S_\epsilon(\omega_1, \omega_2) \quad (\text{C.41})$$

Algunas propiedades de la Función de Densidad Espectral de sucesiones aleatorias reales son las siguientes:

Propiedad	Unidimensional	Bidimensional
Transformada de Fourier	$S(\omega) \leftrightarrow r(n)$	$S(\omega_1, \omega_2) \leftrightarrow r(m, n)$
Real	$S(\omega) = S^*(\omega)$	$S(\omega_1, \omega_2) = S^*(\omega_1, \omega_2)$
Par	$S(\omega) = S(-\omega)$	$S(\omega_1, \omega_2) = S(-\omega_1, -\omega_2)$
No negativa	$S(\omega) \geq 0, \forall \omega$	$S(\omega_1, \omega_2) \geq 0, \forall \omega_1, \omega_2$
Salida de sistemas lineales	$S_u(\omega) = H(\omega) ^2 S_\epsilon(\omega)$	$S_u(\omega_1, \omega_2) = H(\omega_1, \omega_2) ^2 S_\epsilon(\omega_1, \omega_2)$
Separabilidad		$S(\omega_1, \omega_2) = S_1(\omega_1)S_2(\omega_2)$ si $r(m, n) = r_1(m)r_2(n)$

Ejemplo. 9 La función de covarianza de un campo estacionario de ruido blanco viene dada por $r(m, n) = \sigma^2 \delta_{m,n}$. Su Función de Densidad Espectral es constante

$$S(\omega_1, \omega_2) = \sigma^2 \sum_m \sum_n \delta_{m,n} \exp[-j(2\pi\omega_1 m + 2\pi\omega_2 n)] = \sigma^2$$

Bibliografía

- [1] R.A. Hummel A. Rosenfeld and S.W. Zucker. Scene labelling by relaxation techniques. *I.E.E.E. Trans. on Systems, Man and Cybernetics*, 6:420–433, 1976.
- [2] I.E. Abdou and W.K. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detector. *Proceedings of the I.E.E.E.*, 67:753–763, 1979.
- [3] S.M. Bhandarkar and M.Suk. Qualitative features and the generalized Hough transform. *Pattern Recognition*, 25(9):987–1006, 1992.
- [4] M. Boden. *Artificial Intelligence and Natural Man*. New York: Basic Books, 1977.
- [5] V. Bruce and P.R. Green. *Percepción Visual*. Paidós, 1994.
- [6] J.F. Canny. A computational approach to edge detection. *I.E.E.E. Trans. on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.
- [7] J. Casas. *Óptica*. Justiniano Casas Peláez, 1985.
- [8] J.R. Deller. Tom, dick, and mary discover the dft. *IEEE Signal Processing Magazine*, 11(2):36–50, 1994.
- [9] R.O. Duda and P.E. Hart. Use of the Hough transform to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [10] J.P. Frisby. *Del Ojo a la Visión*. Alianza Psicología, 1987.
- [11] J.J. Gibson. *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979.
- [12] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison Wesley, 1987.

- [13] R. C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison Wesley, 1992.
- [14] R.L. Gregory. *Eye and Brain: The Psychology of seeing*. Third Edition, McGraw-Hill Book Co., New York, 1978.
- [15] E.R. Hancock and J. Kittler. Edge labelling using dictionary-based relaxation. *I.E.E.E. Trans on Pattern Analysis and Machine Intelligence*, 12:165–181, 1990.
- [16] A.R. Hanson and E.M. Riseman. Segmentation of natural scenes. In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, pages 129–164. Academic Press, 1978.
- [17] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision. Vol I*. Addison-Wesley, 1992.
- [18] B. K. P. Horn. *Robot Vision*. M.I.T. Press, 1986.
- [19] P.V.C. Hough. *Methods and Means for Recognizing Complex Patterns*. U.S. Patent 3069654, 1962.
- [20] G.W. Humphreys and V. Bruce. *Visual Cognition*. Lawrence Erlbaum Associates, 1989.
- [21] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [22] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.
- [23] B. Julesz. Texture and visual perception. *Scientific American*, 212:38–48, 1965.
- [24] B. Julesz. Experiments in the visual perception of textures. *Scientific American*, 232:34–43, 1975.
- [25] B. Julesz and J. Miller. Independent spatial-frequency-tuned channels in binocular fusion and rivalry. *Perception*, 4:125–143, 1975.
- [26] A.K. Katsaggelos. *Digital Image Restoration*. Springer Verlag. Cap 1., 1991.
- [27] R.L. Lagendijk and J. Biemond. *Iterative Identification and Restoration of Images*. Kluwer Academic Press, 1991.

- [28] D. Marr. *Vision*. W.H. Freeman and Company, 1982.
- [29] D. Marr and E. Hildreth. Theory of edge detection. *proceedings of the Royal Society of London, Series B*, 207:187–216, 1980.
- [30] D. Marr and H.K. Nishihara. A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B200:269–294, 1978.
- [31] D. Marr and S. Poggio. Directional selectivity and its use in early vision processing. *Proceeding of the Royal Society of London, Series B*, 211:151–180, 1981.
- [32] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.
- [33] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceeding of the Royal Society of London*, B204:151–180, 1979.
- [34] G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.
- [35] D.J. McArthur. Computer vision and perceptual psychology. *Psychological Bulletin*, 92:283–309, 1982.
- [36] M.J. Morgan. Computational theories of vision. *Quarterly Journal of Experimental Psychology*, 36A:157–165, 1984.
- [37] J.D. Murray and W. vanRyper. *Encyclopedia of Graphics File Formats*. O'Reily and Associates, INC, 1996.
- [38] V.S. Nalwa. *A Guided Tour to Computer Vision*. Addison-Wesley, 1993.
- [39] J.K. Parker. *Practical Computer Vision using C*. John Wiley, 1994.
- [40] A.K.C. Wong P.K. Sahoo, S. Soltani and Y.C. Chen. Survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.
- [41] J.M. Prager. Extracting and labelling boundary segments in natural scenes. *I.E.E.E. Trans on Pattern Analysis and Machine Intelligence*, 2:16–27, 1980.
- [42] P.V.C. A method and means for recognizing complex patterns. U.S. Patent N. 3,069,654, 1962.

- [43] R. Kasturi R. Jain and B.G. Schunck. *Machine Vision*. 1996.
- [44] K.R. Rao and J.J. Hwang. *Techniques & Standards for Image, Video & Audio Coding*. Prentice Hall, 1996.
- [45] E.M. Riseman and M.A. Arbib. Computational techniques in the visual segmentation of static scenes. *Computer Graphics and Image Processing*, 6:221–276, 1977.
- [46] A. Rosenfeld and A.C. Kak. *Digital Picture Processing. Vol. 1, 2*. Academic Press, 1982.
- [47] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [48] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman & Hall Computing, 1993.
- [49] S. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, 4:104–119, 1975.
- [50] R.J. Watt. *Visual processing: computational, psychophysical and cognitive reserach*. Lawrence Erlbaum Associates Ltd, 1988.
- [51] W.T. Vettering W.H. Press, S.A Teukolsky and B.P. Flannery. *Numerical recipes in C*. Cambridge University Press, 1992.
- [52] S.W. Zucker. Relaxation labelling, local ambiguity, and low level vision. In C.H. Chen, editor, *Pattern Recognition and Artificial Intelligence*. Academic Press, 1976.