

Índice General

1 Imagen	6
1.1 Procesamiento y análisis de imágenes	6
1.1.1 Imagen	6
1.2 Representación matemática de una imagen digital	7
1.3 Muestreo y cuantización	7
1.3.1 Cambio del número de pixeles por muestra	8
1.3.2 Número de bits por pixel	8
1.4 Vecindad de un pixel	10
1.5 Conectividad	10
2 Formatos de imágenes	12
2.1 Formato de archivo de imágenes BITMAP (BMP o DIB)	12
2.1.1 Estructura general	12
2.2 Referencias	17
3 Transformada de Fourier	18
3.1 Introducción	18
3.2 Transformada de Fourier	18
4 Mejora de imágenes	22
4.1 Histograma	23
4.2 Brillo y contraste	24
4.2.1 Brillo	24
4.2.2 Contraste	24
4.3 Métodos basados en los pixeles (punto a punto)	25
4.3.1 Negativo de una imagen	25
4.3.2 Mejora del contraste	25
4.3.3 Compresión del rango dinámico	26
4.3.4 Realce de rangos de intensidad	26
4.4 Pretratamiento de imágenes	26
4.5 Modificación del histograma	26
4.5.1 Condiciones para la aplicación de la modificación del histograma	27
4.5.2 Ecualización o linealización	27
4.5.3 Especificación del histograma	29
4.6 Transformación local del histograma	30
4.7 Imagen promedio	31
4.8 Substracción de imágenes	31
4.9 Filtrado espacial	32
4.9.1 Filtro pasabajos	33
4.9.2 Filtro gaussiano	35
4.9.3 Filtros de realce	36
4.9.4 Filtro pasaaltos	36

4.9.5	Filtro de realce	37
4.9.6	Filtro del máximo	38
4.9.7	Filtro del mínimo	39
4.9.8	Filtro mediana	39
4.9.9	Filtro de realce en relieve	39
4.9.10	Detección de bordes	40
4.9.11	Máscaras de gradiente	43
4.9.12	Umbralización del gradiente	51
4.9.13	Comparación entre las máscaras de gradiente	51
4.9.14	Laplaciano	52
5	Segmentación por bordes	55
5.1	Segmentación basada en la obtención de bordes	55
5.2	Métodos avanzados de segmentación de imágenes por bordes	56
5.2.1	Operador de Marr y Hildreth	57
5.2.2	Aproximación de Canny	61
5.2.3	Aproximación de Deriche	66
5.2.4	Aproximación de Shen y Castan	72
5.2.5	Anotaciones	75
6	Selección de los pixeles del contorno	76
6.1	Selección a partir del gradiente	76
6.1.1	Umbralización simple	76
6.1.2	Umbralización por histeresis de los máximos locales del gradiente	76
6.2	Selección a partir del laplaciano	78
6.2.1	Supresión de cruces por cero falsos	79
6.2.2	Umbralización por gradiente adaptivo	79
7	Segmentación por regiones	80
7.1	Generalidades	80
7.2	Métodos para el cálculo de un umbral óptimo	82
7.2.1	Binarización global	82
7.3	Método de crecimiento por regiones	90
7.4	Método de dividir y mezclar (splitting & merging)	91
7.5	Etiquetado	92
8	Segmentación por multiresolución	94
9	Detección de primitivas	95
9.1	Transformada de Hough	95
9.1.1	Transformada de Hough de líneas rectas	95
9.1.2	Transformada de Hough de otras primitivas	97
10	Descriptores de región	99
11	Modelos de contorno activo: Snakes	101
11.1	Introducción	101
11.2	Modelos deformables	101
11.2.1	Clasificación	101
11.3	Estructura de los modelos de contorno activo	102
11.3.1	Energía interna	102
11.3.2	Energía elástica	103
11.3.3	Energía de la imagen	104
11.4	Aplicaciones	105

12 Morfología matemática	106
12.1 Elemento estructurante	106
12.2 Erosión	107
12.3 Dilatación	108
12.4 Apertura	109
12.5 Cerrado	110
12.6 Obtención del contorno de un objeto	110
12.6.1 Operador morfológico de gradiente	112
12.6.2 Relleno de regiones mediante dilatación condicional	112
12.6.3 Detección de elementos conectados	113
12.7 Filtro sal y pimienta	113
12.8 Filtro de mayoría negra	114
12.9 Adelgazamiento y esqueletización	114
12.9.1 Obtención del esqueleto mediante la transformación del eje medio	114
12.9.2 Adelgazamiento	115
12.9.3 Detección de líneas	115
12.10 Seguimiento del contorno	116
13 Transformaciones geométricas de una imagen en el espacio bidimensional	117
13.1 Representación de puntos	117
13.2 Cambio de escala	117
13.2.1 Escalización en x	117
13.2.2 Escalización en y	118
13.2.3 Forma general de la escalización	118
13.3 Reflexión	118
13.3.1 Reflexión alrededor del eje x	119
13.3.2 Reflexión alrededor del eje y	119
13.3.3 Reflexión alrededor del origen	119
13.4 Deformación	119
13.5 Rotación	120
13.6 Traslación y coordenadas homogéneas	120
13.7 Rotación alrededor de un punto arbitrario	121
13.8 Técnicas de mapeo	122
14 Transformaciones geométricas de una imagen en el espacio tridimensional	123
14.1 Escalización	124
14.2 Deformación	124
14.3 Rotación	124
14.4 Reflexión alrededor de una línea arbitraria	124
14.5 Proyección	125
14.5.1 Proyección ortográfica	125
14.5.2 Proyección ortogonal	126
14.5.3 Transformación en perspectiva	126
15 Composición de imágenes	128
15.1 Composición sencilla	128
15.2 Composite	128
15.3 Canal alfa	129
15.4 Algebra del composite	130
15.5 Realización del composite	132
15.6 Operadores del composite	132
15.6.1 Factor de desvanecimiento	132
15.6.2 Factor de oscurecimiento	132
15.6.3 Factor de opacidad	133

ÍNDICE GENERAL

	4
15.6.4 Operador de adición	133
16 Transformada de Karhunen-Loève	134
17 Fusión de imágenes mediante el uso de sinogramas	137
17.1 Introducción	137
17.2 Métodos de proyección	137
17.2.1 Proyección de Máxima Intensidad	138
17.2.2 Promedio	138
17.3 Sinograma o transformada de Radón	138
17.4 Correlación	140
17.5 Procedimientos	140
18 Metamorfosis	143
18.1 Historia	143
18.2 Técnicas de metamorfosis	143
18.2.1 Deformación por mallado	144
18.2.2 Metamorfosis de campo	144
18.3 Metamorfosis de campo	144
18.3.1 Transformación de un par de líneas paralelas	144

Introducción

Con el nacimiento del computador y su posterior desarrollo, surgió la necesidad de emplearlo como herramienta para representar la realidad por medio de imágenes, y aún más, que la misma máquina fuera capaz de identificar lo que ellas representan.

Un primer problema para resolver fue lograr que una imagen analógica o continua pudiera ser cargada en el computador. Para ello se diseñaron diferentes técnicas de adquisición. Una vez digitalizada la imagen, se hace necesario procesar los datos para lograr extraer la mayor cantidad de información posible con la mejor calidad o simplemente modificarlos con fines técnicos o artísticos. La difusión de estos métodos es limitada. La rápida y continua evolución en este campo hace de los textos existentes un material limitado y poco actualizado en cuanto a técnicas y procedimientos. Este campo específico requiere además una alta formación práctica.

El tratamiento de imágenes es una herramienta de importancia relevante en muchas aplicaciones de la vida cotidiana. Se encuentran aplicaciones de esta rama en la medicina, el cine, la aviación, el reconocimiento de terrenos, las imágenes satelitales y en fin en numerosos campos que requieren el mejoramiento de las imágenes.

En Colombia, el procesamiento de imágenes y la computación gráfica, representan un campo poco estudiado y hacia el cual no se dirigen muchos esfuerzos salvo algunas excepciones. En particular, en la Universidad Nacional de Colombia se observó la necesidad de tratar temas relacionados con la computación gráfica, el tratamiento de imágenes y sonido y el software multimedial, que sin duda son determinantes dentro del desarrollo tecnológico que exige la época; de esta manera hace unos pocos años se ha incorporado el área de la computación gráfica dentro del pensum académico de la Facultad de Ingeniería de Sistemas y es así como se pretende en estas clases incentivar la investigación y el estudio de los múltiples aspectos que encierra el área mencionada.

Este texto ha sido elaborado con el fin de cubrir la necesidad, presente en el curso de imágenes, de un texto guía que abarque tanto los temas básicos de un curso de tratamiento de imágenes, como los últimos avances en este campo, con un alto contenido práctico.

Este libro está orientado a estudiantes del curso de procesamiento de imágenes en carreras de ingeniería.

El texto comienza con una introducción a las imágenes empleadas en el procesamiento y técnicas de muy bajo nivel para ir avanzando progresivamente hacia técnicas de mayor complejidad.

Los primeros capítulos presentan los temas vistos a lo largo del curso básico. Los últimos presentan una serie de nuevos temas estudiados como investigación final.

Quiero agradecer la colaboración de Carlos Galeano, Luis Gustavo García, Marcela Gonzalez, Eustorgio Reina y Eduard Sierra. Igualmente a los estudiantes de mis cursos de imágenes que me motivaron para escribir este libro.

Manuel Guillermo Forero V.

Capítulo 1

Imagen

1.1 Procesamiento y análisis de imágenes

El procesamiento digital de imágenes nació desde el origen mismo de los computadores digitales, cuando se hizo notoria la gran potencialidad de los computadores en el tratamiento de las señales visuales observables por el ser humano. El primer problema presentado consistió en la determinación de las discontinuidades que aparecen en una imagen, buscando distinguir los objetos de interés del fondo de la imagen. Las investigaciones se centraron inicialmente en el estudio del sistema visual humano, pero dada su enorme complejidad y el escaso progreso en las teorías y algoritmos que explicaban su funcionamiento y que podrían ser aplicables al procesamiento de imágenes, se buscó una solución más directa con miras a emplearlas sobre computadores.

Los primeros trabajos se centraron en la transmisión y codificación de imágenes. Más adelante, con la aparición de las imágenes satelitales nació la necesidad de desarrollar técnicas más eficientes de transmisión de imágenes y nuevos métodos de realce y restauración que permitieran el mejoramiento en su calidad[42].

Con el tiempo, aparece una clara especialización en las técnicas. Por un lado aparecen los métodos de procesamiento de imágenes, desarrollados para la captura, codificación y representación de éstas. Por el otro, los métodos de análisis empleados para extraer la información contenida en la imagen.

1.1.1 Imagen

Se puede definir una imagen bidimensional estática como una representación planar de una escena o de un objeto situado, en general, dentro de un espacio tridimensional. Se obtiene cada vez que un captor es sensibilizado por los rayos que han interactuado con los objetos. El modelo del proceso puede ser visto desde diferentes puntos de vista:

- Modelo abstracto.
- Modelo geométrico de la transformación de una escena 3D sobre un plano 2D.
- Modelo óptico.
- Modelo espacial.
- Modelo del comportamiento espectral.
- Modelo digital.

Este último aparece con la necesidad de utilizar herramientas computacionales en el procesamiento de imágenes. En este caso, una imagen continua o analógica, que representa un objeto o una escena dada, al ser modelizada mediante una digitalización se traduce en una serie de muestras igualmente espaciadas formando un arreglo bidimensional, donde a cada elemento corresponde una amplitud asociada a un color, intensidad luminosa o nivel de gris.

1.2 Representación matemática de una imagen digital

La función representativa de una imagen puede ser de diferentes tipos:

- De valores reales con dos argumentos para una película fotográfica.
- De valores discretos para una imagen digital.

Dentro del procesamiento de imágenes se trabaja con imágenes digitalizadas. En éstas los valores espaciales y de intensidad han sido discretizados. En este caso, las imágenes pueden ser vistas como un arreglo bidimensional de $N \times M$ elementos llamados pixeles. El término pixel proviene de la contracción de las palabras inglesas picture y element, para significar que es la unidad básica de representación de una imagen digital. En el caso de una imagen en niveles de gris, también llamada monocromática, puede ser definida como un conjunto de pixeles, donde cada pixel tiene asignado un nivel de gris característico q correspondiente a la luminosidad o intensidad en el punto. q varía entre 0 y $L-1$, siendo L el número de niveles de gris disponibles en la imagen.

Generalmente en el caso monocromático una imagen bidimensional es una función de dos variables $q(x,y)$ representando la luminosidad en el pixel de coordenadas (x,y) .

$$I = \{q(x,y) \mid 0 \leq x < M, 0 \leq y < N\}$$

El número de bits B requeridos para representar la imagen es:

$$B = M.N.b$$

donde:

- M : número de líneas de la imagen.
- N : número de columnas.
- b : número de bits por pixel.

Para una imagen multiespectral, q es una función vectorial de n componentes. Por ejemplo, en el caso de una imagen en color, donde el color compuesto esté expresado como una combinación de las luces básicas rojo (r), verde (v) y azul (a).

$$q(x,y) = (q_r(x,y), q_v(x,y), q_a(x,y))$$

donde q_r , q_v y q_a corresponden a los niveles de rojo, verde y azul respectivamente.

1.3 Muestreo y cuantización

Para representar una imagen en un computador ésta debe ser discretizada. Así, la modelización de una imagen obtenida de una escena original continua o análoga, se obtiene mediante su digitalización en espacio e intensidad o luminosidad para obtener una imagen discreta. La discretización puede ser vista como la subdivisión de la imagen en pequeñas unidades de representación o pixeles, cada una de los cuales tiene un valor de gris o color asociado. Al capturar una imagen se presentan pérdidas de información debido a:

- La digitalización espacial como consecuencia del muestreo de la imagen dada la naturaleza discreta de los pixeles.
- La digitalización en amplitud o cuantización dado el rango limitado de los valores de intensidad luminosa que puede tomar un pixel.

1.3.1 Cambio del número de pixeles por muestra

El muestreo de una imagen equivale a la discretización de las coordenadas espaciales de ésta y tiene por efecto la reducción de su resolución espacial. En la figura 1.1 se observa el efecto de la adquisición de la misma imagen con diferentes resoluciones de adquisición. Se aprecia que a medida que disminuye el número de muestras por imagen se presenta una mayor pérdida de información y un aumento en el ruido introducido en forma de patrones rectangulares en la imagen.



Figura 1.1: Efectos de la cuantización de la imagen.

1.3.2 Número de bits por pixel

La discretización de la amplitud de un punto, equivale a la cuantificación de su nivel de gris. La figura 1.2 muestra los efectos de la reducción del número de bits por pixel empleados, lo cual implica una reducción en los niveles de gris usados por imagen y por ende una menor calidad de representación. El problema de la cuantización se origina debido a la imposibilidad de tener un rango infinito de valores para medir la intensidad de los pixeles de una imagen.



Figura 1.2: Imágenes obtenidas variando el número de niveles de gris sobre el cual se muestrea una imagen.

No existe un criterio que permita escoger el número óptimo de bits por pixel y de pixeles por muestra para digitalizar una imagen. Sin embargo, a partir de diferentes estudios, se ha llegado a la conclusión de que por encima de ciertos valores de umbral, una persona no experimenta una mejora visual en la calidad de la imagen, pero por debajo de estos si observa una pérdida. De todas maneras, para el procesamiento digital de las imágenes, en casos de reconocimiento de formas, mejora, análisis e interpretación, etc. es mucho mejor contar con imágenes con la mejor calidad y resolución posibles.

1.4 Vecindad de un pixel

Un pixel $q(x,y)$ tiene cuatro u ocho vecinos dependiendo del criterio usado. Si se definen como sus vecinos los pixeles $q(x, y - 1)$, $q(x - 1, y)$, $q(x + 1, y)$ y $q(x, y + 1)$, mostrados en la figura 1.3.a; este conjunto notado $N_4(q)$, es llamado 4-vecinos de q . Los pixeles $q(x - 1, y - 1)$, $q(x + 1, y - 1)$, $q(x - 1, y + 1)$ y $q(x + 1, y + 1)$, mostrados en la figura 1.3.b constituyen los vecinos diagonales de q llamados $N_D(q)$ y la reunión en un solo conjunto de $N_4(q)$ y $N_D(q)$ es llamado los 8-vecinos de q , denotado $N_8(q)$ y mostrados en la figura 1.3.c.

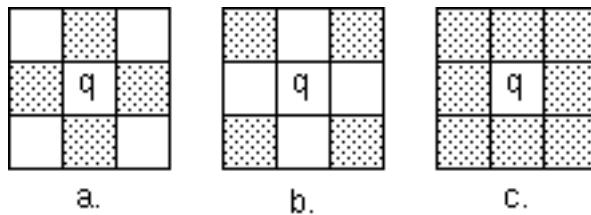


Figura 1.3: En sombreado se encuentran según su definición de los vecinos del pixel q . a. 4-vecinos. b. D-vecinos. c. 8-vecinos.

1.5 Conectividad

La conectividad es una propiedad topológica empleado en el procesamiento de imágenes con el fin de determinar las fronteras de los objetos y separar las diferentes zonas que aparecen en la imagen.

Se dice que dos pixeles están conectados si estos son adyacentes de acuerdo a algún criterio y de acuerdo a otro sus niveles de gris son similares.

Por ejemplo, si en la figura 1.4 se desean conocer los pixeles conectados al d , tomando como criterios de adyacencia a sus 4-vecinos y de similaridad a los que tengan su mismo nivel de gris, se encuentra que solo el pixel c los cumple y por tanto se dice que c está conectado a d .



Figura 1.4: Definición de pixeles conectados.

Dados dos pixeles q_1 y q_2 , donde el nivel de gris de q_2 está contenido en un intervalo de valores $[q_i, q_f]$ considerados como similares a q_1 , pueden definirse tres tipos de conectividad:

- 4-conectividad: q_1 y q_2 están conectados en conectividad 4 si q_2 es 4-vecino de q_1 , es decir, $q_2 \in N_4(q_1)$.
- 8-conectividad: q_1 y q_2 están conectados en conectividad 8 si q_2 es 8-vecino de q_1 , es decir, $q_2 \in N_8(q_1)$.
- Conectividad mixta : Esta forma de conectividad resulta de una modificación de la 8-conectividad y permite la eliminación de conexiones múltiples, las cuales son en algunos casos fuentes de error [3][20]. La figura 1.5 muestra un ejemplo. Se dice que los pixeles q_1 y q_2 están conectados en conectividad mixta si cumplen una de las siguientes dos condiciones:

1. $q_2 \in N_4(q_1)$.
2. $q_2 \in N_D(q_1)$ y el conjunto $N_4(q_1) \cap C_D(q_2)$ es vacio.

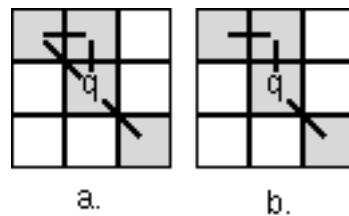


Figura 1.5: Definición de conectividad mixta. Las líneas negras indican la conectividad entre pixeles. a. Pixeles conectados en 8-conectividad al pixel q . b. Pixeles conectados en conectividad mixta al pixel q .

Capítulo 2

Formatos de imágenes

Antes de comenzar a tratar las imágenes, es conveniente comprender como éstas son almacenadas en el computador en lo que se denominan archivos gráficos. Dentro de ellos se estudia uno de los más sencillos y más ampliamente usados: el BMP o DIB; se describen las estructuras básicas que lo componen y la descripción de cada uno de sus miembros.

2.1 Formato de archivo de imágenes BITMAP (BMP o DIB)

El formato de archivos Bitmap (BMP o DIB, del inglés Device Independence Bitmap) es el estándar empleado por Microsoft para el manejo de imágenes sobre el ambiente Windows y es uno de los más utilizados, debido a la simplicidad de su estructura y facilidad de manipulación. Este formato permite trabajar con imágenes independiente del dispositivo o computador sobre el cual se este trabajando; para lograrlo, el archivo especifica el color de cada pixel independiente de la forma en que el dispositivo lo representa en la pantalla.

Los tipos de datos utilizados por las estructuras que describen este formato son los siguientes:

Tipo de dato	Longitud
BYTE	8 bits
WORD	16 bits
DWORD	32 bits
LONG	32 bits

2.1.1 Estructura general

Un archivo BMP está constituido por 4 estructuras básicas como se ilustra en la figura 2.1.

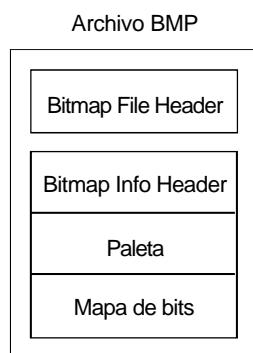


Figura 2.1: Estructura del formato BMP.

A. Estructura BITMAPFILEHEADER

Esta estructura consta de un conjunto de datos que se encuentra encabezando el archivo, tiene un tamaño de 14 bytes y en ella se encuentra la información relevante para la interpretación del archivo. Está compuesta por los siguientes datos (figura 2.2):

WORD bfType: Especifica el tipo de archivo, consta de la cadena de caracteres 'BM', que expresada en hexadecimales corresponde a 4D42H ó 19778 en decimal.

DWORD bfSize: Especifica el tamaño de la imagen, aunque generalmente no contiene un valor confiable y es preferible no utilizarlo.

WORD bfReserved1: Reservado; debe ser cero.

WORD bfReserved2: Reservado; debe ser cero.

DWORD bfOffBits: Especifica la distancia en bytes que separa el inicio del archivo del comienzo de la imagen almacenada en éste.

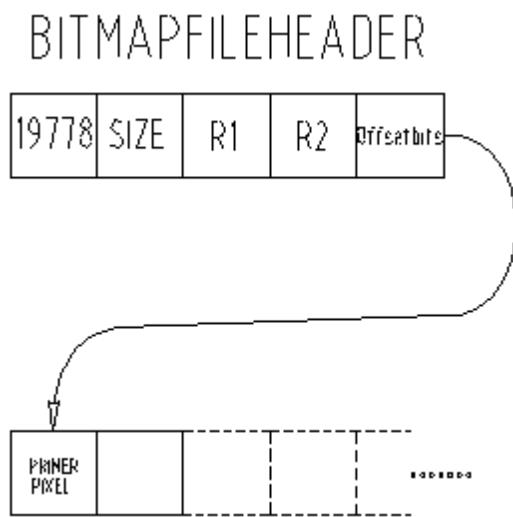


Figura 2.2: Estructura BITMAP FILE HEADER.

B. Estructura BITMAPINFOHEADER

Esta segunda estructura contiene información general acerca de la imagen, también llamada bitmap o mapa de bits.

DWORD biSize: Contiene el tamaño en bytes de esta estructura. Para versiones anteriores a Windows 3.1 este es de 12 y para versiones posteriores es de 40.

LONG biWidth: Contiene el ancho de la imagen expresado en pixeles.

LONG biHeight: Especifica el alto de la imagen expresada en pixeles.

WORD biPlanes: Contiene el número de planos en el bitmap. Su valor debe ser 1.

El número de planos se refiere a la forma en que los datos están organizados en el bitmap, cuando se trabaja con imágenes en color, en niveles de gris o que incluyen un canal de transparencia (canal α) como se ve en el capítulo de composición de imágenes. Esta información no es requerida en imágenes en nivel de gris, si esta no incluye un canal de transparencia. Un bitmap puede contener la información del color en dos formas diferentes. En la primera, llamada en inglés mesh, la información acerca de cada pixel es almacenada en forma contigua

(ver figura 2.3), es decir, los tres diferentes valores de los componentes de color para cada pixel (rojo, verde y azul) están guardados en el archivo uno a continuación del otro. En la segunda mostrada en la figura 2.4, llamada configuración planar, cada componente de color de cada pixel es guardada en forma contigua formando tres planos, uno para cada color. Es decir, todos los valores de color rojo de cada pixel son guardados uno detrás de otro, seguido por los verdes y finalmente por los azules.

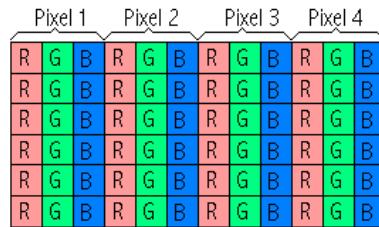


Figura 2.3: Bitmap de colores guardados en forma no planar, llamada en inglés mesh.

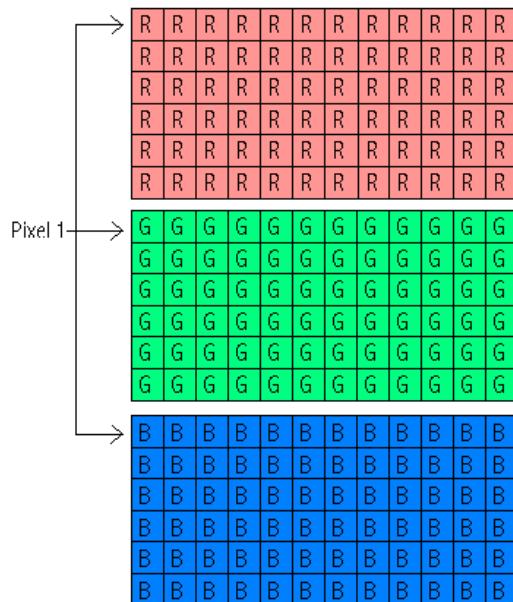


Figura 2.4: Bitmap de colores guardados en forma planar.

WORD biBitCount: Especifica el número de bits utilizados para representar cada pixel. Su valor puede ser 1, 4, 8 o 24.

DWORD biCompression: Especifica el tipo de compresión del bitmap. Puede tomar uno de los siguientes valores:

Valor	Significado
BI_RGB = 0	Bitmap sin compresión.
BI_RLE8 = 1	Especifica el formato de codificación para bitmaps con 8 bits por pixel.
BI_RLE4 = 2	Especifica el formato de codificación para bitmaps con 4 bits por pixel.

En general, las imágenes en formato bitmap no se encuentran comprimidas.

DWORD biSizeImage: Especifica el tamaño, en bytes, de la imagen. Es válido dejar en cero este miembro si el bitmap no está comprimido.

LONG biXPelsPerMeter: Contiene la resolución horizontal de la imagen, en pixeles por metro del dispositivo origen para el bitmap. Con este valor y el siguiente se puede determinar el tamaño real de un objeto visualizado en la imagen.

LONG biYPelsPerMeter: Contiene la resolución vertical de la imagen, en pixeles por metro del dispositivo origen para el bitmap.

DWORD biClrUsed: Especifica el número de índices dentro de la tabla de colores que contiene el archivo que en realidad es utilizada para representar la imagen, es decir, indica el número de índices de color presentes en la tabla de colores usados por el bitmap. Si este valor es cero, el bitmap usa el máximo número de colores correspondientes al valor del miembro biBitCount. Si biClrUsed no es cero, significa que el número de colores disponibles en la paleta está definido en el valor miembro biBitCount, siempre y cuando este valor sea menor a 24. Si biBitCount es igual a 24, biClrUsed indica que no hay una paleta de colores y se podrá emplear toda la gama de colores disponible y representable en 24 bits.

Si el bitmap está comprimido, el miembro biClrsed debe estar en cero o con el valor del tamaño de la paleta de colores.

DWORD biClrImportant: Especifica el número de colores de la paleta que se consideran importantes para desplegar el bitmap, para ello los colores en paleta deben aparecer en orden de importancia. Si este valor es cero, todos los colores se consideran importantes. Este valor es usado cuando una imagen debe ser desplegada en un monitor que no permite presentar un gran número de colores.

C. Paleta

La paleta es una tabla que contiene la lista de colores empleados al construir la imagen. En caso de que el número de colores empleados para representar la imagen sea correspondiente a 24 bits o más, no se incluye la paleta dentro del archivo. Esta se representa por un arreglo de estructuras RGBQUAD que define los colores en el bitmap. Una estructura RGBQUAD consta de los siguientes miembros:

BYTE rgbBlue: Especifica la intensidad de azul en el color.

BYTE rgbGreen: Especifica la intensidad de verde en el color.

BYTE rgbRed: Especifica la intensidad de rojo en el color.

BYTE rgbReserved: Este es un campo reservado y debe contener el valor cero.

Como se dijo antes, el miembro biBitCount de la estructura BITMAPINFOHEADER, determina el número de bits que define cada pixel y el número máximo de colores en el bitmap. Este miembro puede tener los siguientes valores:

Valor	Significado
1	El bitmap es monocromático (2^1 colores disponibles) y por tanto, la paleta debe contener 2 entradas. Cada bit en el arreglo del bitmap representa un pixel. Si el bit es cero, el pixel es mostrado con el color de la primera entrada en la paleta. Si el bit es uno, el pixel contiene el color correspondiente a la segunda entrada en la paleta.
4	El bitmap tiene un máximo de 2^4 colores y la paleta contiene 16 entradas. Cada pixel en el bitmap es representado por cuatro bits dentro de la tabla de colores. Por ejemplo, si el primer byte en el bitmap es $1F$ h el byte representa dos pixeles. El primer pixel contiene el color en la segunda entrada en la tabla, y el segundo pixel contiene el color de la decimosexta entrada de la tabla.
8	El bitmap tiene un máximo de 2^8 colores y por consiguiente la paleta consta de 256 entradas. En este caso, cada byte dentro del arreglo representa un pixel.
24	El bitmap tiene un máximo de 2^{24} (más de 16 millones) colores. En este caso, el archivo no contiene ninguna paleta y cada secuencia de tres bytes en el arreglo del bitmap representa las intensidades relativas de rojo, verde y azul del pixel respectivo.

D. Mapa de bits

El mapa de bits (en inglés bitmap) constituye la imagen propiamente dicha. Puede interpretarse como una matriz, donde el número de filas es igual a la altura de la imagen en pixeles y el número de columnas corresponde al ancho de la misma. Las líneas de la imagen se encuentran organizadas en forma opuesta a como se hallan en el bitmap. El primer pixel hallado en el bitmap corresponde al de la esquina inferior izquierda de la imagen y los demás pixeles se leen siguiendo la secuencia mostrada en la figura 2.5, es decir de izquierda a derecha y de abajo hacia arriba. En caso de que el campo biHeight de la estructura sea negativo, la correspondencia entre el bitmap y la imagen es directa; es decir, el primer dato se encuentra ubicado en la esquina superior izquierda y los datos siguientes se leen de izquierda a derecha y de arriba hacia abajo.

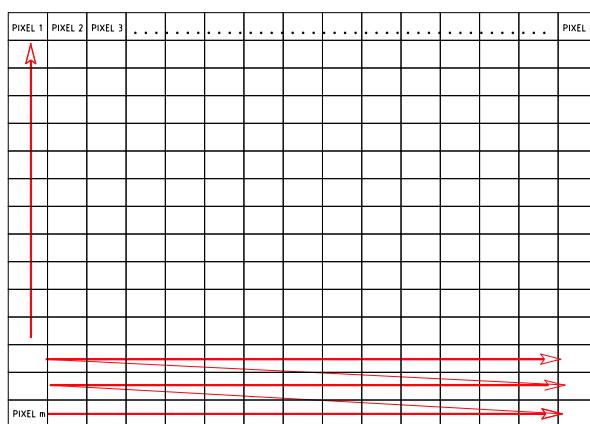


Figura 2.5: Secuencia de almacenamiento del mapa de bits dentro de un archivo BMP típico.

Es importante anotar que el ancho del bitmap debe ser un múltiplo de 4, pues en esta forma se facilita el despliegue de la imagen en la pantalla. En caso de que el ancho no sea un múltiplo de 4, este se fuerza a serlo, adicionando el número de bits necesario hasta para ello. Por lo tanto, es posible que el campo biSizeImage pueda no corresponder a biHeightxbiWidth.

2.2 Referencias

Varios libros se han escrito sobre el tema de formatos de archivos de imágenes, se sugieren las siguientes referencias:

Luse M. Bitmapped graphics programing in C++. Addison Wesley.1993.

Lindley C. A. Practical image processing in C. John Wiley & Sons. 1991.

Si se requiere una información adicional sobre el formato BMP, esta se encuentra en la ayuda en línea de varias aplicaciones como Microsoft Visual C++ o Borland C++Builder y el libro:

Thompson N. Técnicas de animación con Windows 95 y Windows NT. Mc Graw Hill. 1995.

Capítulo 3

Transformada de Fourier

3.1 Introducción

La transformada de Fourier en una dimensión tiene diversas aplicaciones. Dado que es posible generalizarla a más dimensiones, es posible aplicarla en el procesamiento de imágenes, permitiendo por ejemplo, el aplicar filtros directamente en el dominio de la frecuencia y ver el resultado al regresar al dominio espacial.

Quizas su mayor limitante es el tiempo de cálculo requerido para obtener la transformada. Implementando el algoritmo directamente a partir de la definición de la transformada directa de Fourier (TDF) requiere de un número elevado de operaciones. Sin embargo, basandose en algunas propiedades de la TDF y de algunas técnicas de programación se logra construir un algoritmo más eficiente denominado transformada rápida de Fourier (TRF) que permite reducir considerablemente el tiempo de cálculo.

3.2 Transformada de Fourier

El proceso mediante el cual se aplica un filtro espacial h a una imagen q , se expresa por una operación de convolución:

$$q(x, y) * h(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} q(a, b)h(x - a, y - b)dadb$$

donde x, y son las coordenadas del pixel y a, b son variables artificiales para el desplazamiento de la función a convolucionar.

La transformada de Fourier permite convertir una función que se encuentra en el dominio del tiempo o del espacio al dominio de la frecuencia. Esta representación es muy importante en diversas áreas como por ejemplo en el análisis de señales, comunicaciones, etc.

En el procesamiento de imágenes, la transformada de Fourier $F(u, v)$ de una imagen $q(x, y)$ puede interpretarse como la variación de los niveles de gris entre un pixel y sus vecinos. Sobre esta función pueden realizarse las operaciones de filtrado conocidas en el dominio de la frecuencia y aplicadas en el procesamiento de señales: pasabajos, pasaaltos, las cuales se reducen a simple multiplicaciones entre el filtro y la señal original.

En el caso de las imágenes la transformada de Fourier es un proceso por medio del cual, se puede obtener información en el dominio de la frecuencia de una señal que está originalmente en el dominio del espacio. Esta transformada es definida para señales continuas en el espacio, pero para su manejo en el computador se define su equivalente discreto, dado por [3][20]:

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

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi j(\frac{ux}{M} + \frac{vy}{N})} \quad u = 0, \dots, M-1, v = 0, \dots, N-1 \quad (3.1)$$

donde,

$f(x, y)$: señal discreta en el dominio del espacio.

$F(u, v)$: señal discreta equivalente en el dominio de la frecuencia.

x, y : coordenadas espaciales.

u, v : índices de frecuencia.

M, N : número de filas y columnas de la imagen respectivamente.

Para obtener la transformada de Fourier de una imagen puede aplicarse primero la transformada a cada fila o columna y sobre el resultado se aplica la transformada a cada columna o fila, lo cual corresponde a la ecuación 3.1.

Entre las propiedades de la transformada de Fourier se tiene el teorema de convolución dado por:

$$i(x, y) = q(x, y) * h(x, y) \Leftrightarrow I(u, v) = Q(u, v) \cdot H(u, v) \quad (3.2)$$

donde:

el símbolo $*$ representa el operador de convolución,

$i(x, y)$ es la imagen filtrada vista en el dominio del espacio.

$I(u, v)$ es la imagen filtrada vista en el dominio de la frecuencia.

$h(x, y)$ es el filtro visto en el dominio del espacio.

$H(u, v)$ es el filtro visto en el dominio de la frecuencia.

La ecuación 3.2 indica que el proceso de convolución espacial es equivalente a la multiplicación en frecuencia.

La relación de esta importante propiedad con el proceso de filtrado de imágenes radica en que si se tienen la imagen y el filtro en el dominio de la frecuencia, sus componentes se multiplican término a término y se calcula la transformada inversa de la multiplicación obteniendo la imagen filtrada. El proceso se muestra en la figura 3.1.

Una imagen es originalmente una función real, mientras que su transformada es, en general, una función compleja. Así, para visualizar la transformada de Fourier de una señal, se hace uso del espectro de magnitud, definido como:

$$|F(u, v)| = \sqrt{(\Re(F(u, v))^2 + \Im(F(u, v))^2)}$$

donde $\Re(F(u, v))$ y $\Im(F(u, v))$ representan la parte real e imaginaria de $F(u, v)$ respectivamente.

El espectro de magnitud que se obtiene al aplicar la transformada de Fourier tiene el inconveniente de estar centrado en el origen (esquina superior izquierda de la imagen), lo cual dificulta una correcta percepción de frecuencias espaciales altas y bajas. Para facilitar el análisis puede centrarse el espectro en el centro de la imagen, lo cual se logra multiplicando los valores originales por $-x-y$ y calculando la transformada de Fourier sobre este resultado. La figura 3.2 presenta un espectro original, centrado en el origen, y su correspondiente centrado en el centro.

La forma general que presenta un espectro de magnitud centrado en el centro se muestra en la figura 3.3.

En adelante, todos los espectros de magnitud presentados están centrados en el centro de la imagen.

En una imagen los bordes y cambios bruscos en los niveles de gris están relacionados con los componentes de alta frecuencia, mientras factores como el color y la iluminación están relacionados con las bajas frecuencias.

La figura 3.4 presenta un ejemplo de aplicación de un filtro mediante el uso de la transformada de Fourier.

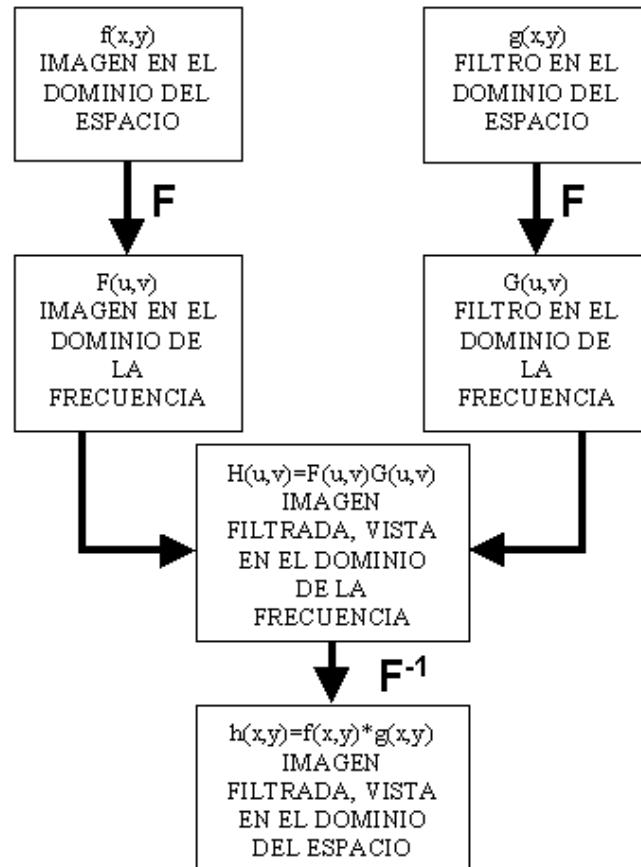


Figura 3.1: Proceso de filtrado usando la transformada de Fourier.

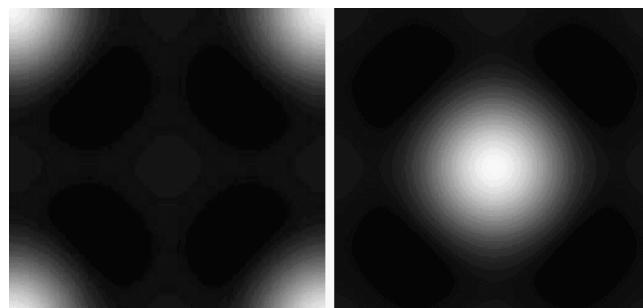


Figura 3.2: Espectro de magnitud de la transformada de Fourier: a. Centrado en el origen. b. Centrado en el centro de la imagen.

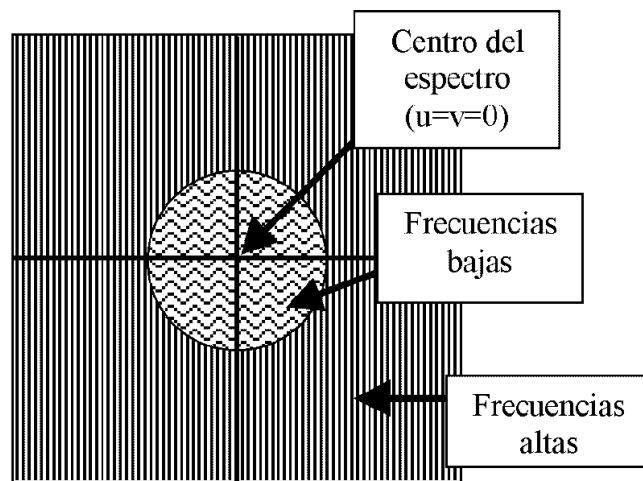


Figura 3.3: Esquema general del espectro de magnitud de una señal bidimensional. Adaptado de [18].

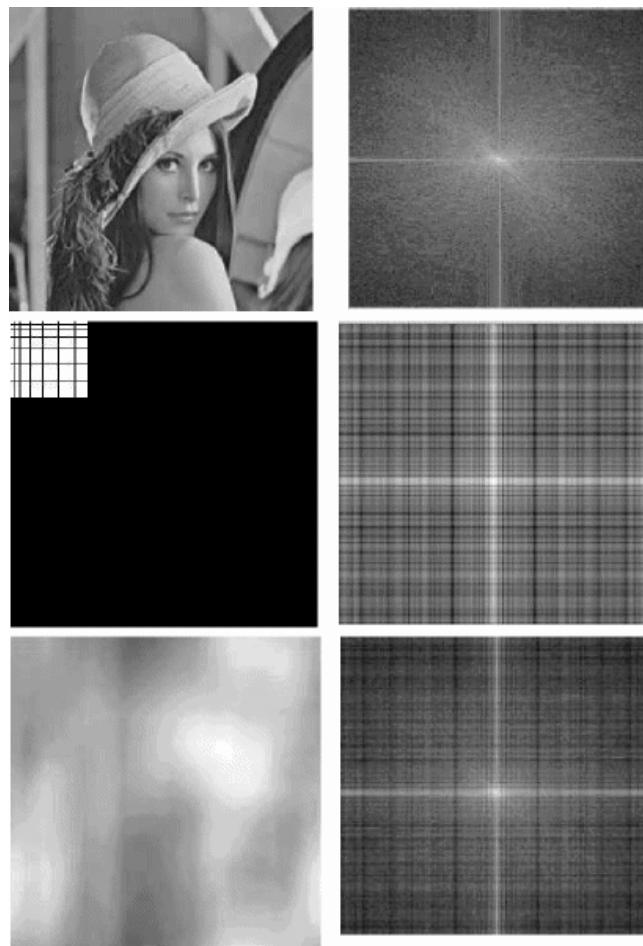


Figura 3.4: Proceso de filtrado de una imagen mediante la transformada de Fourier. Arriba. Imagen original y su espectro de magnitud. Centro:Imagen del filtro a aplicar y su espectro de magnitud. Abajo. Imagen filtrada resultante. Adaptado de [18].

Capítulo 4

Mejora de imágenes

Una vez adquirida la imagen puede ser deseable modificarla, bien sea con el fin de mejorar la representación que hace de la realidad o para resaltar algunas características particulares. Es necesario considerar en forma simultánea las técnicas de mejora y restauración de imágenes, que corresponden a un aumento en la calidad de la imagen. En efecto, en la adquisición de la imagen, dadas las características no ideales del captor y la presencia de ruido, la imagen obtenida es diferente de la imagen deseada (ver figura 4.1).

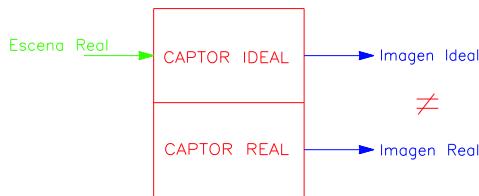


Figura 4.1: Diferencia en la captura entre la imagen real y la imagen ideal.

Varios problemas deben ser considerados:

- Corrección de defectos generados durante la adquisición.
- Estimación de la señal deseada a partir de la observación disponible.
- Reconstrucción de la imagen a partir de datos monodimensionales.
- Compresión de la información.

Así también, algunos criterios deben ser revisados dependiendo de si:

- Las degradaciones son ligeras o importantes.
- Los modelos de las degradaciones son conocidos o no.

No se puede olvidar que la complejidad del pretratamiento ayuda a la robustez de las herramientas de análisis.

Cuando la mejora está destinada a combatir un proceso de degradación conocido o estimado por métodos obtenidos de la teoría de filtraje, se habla más bien de restauración de imágenes.

En esta sección se presentan algunos de los métodos más elementales: modificación de la escala de grises, suavización, etc, que tienen por objetivo devolver una imagen de más fácil comprensión. Estos métodos son esencialmente interactivos, en el sentido de que el sistema de referencia es la visión humana. Se les encuentra generalmente implementados en las consolas de tratamiento, dado su empleo frecuente.

Los diferentes métodos pueden ser clasificados en:

- Métodos basados en el pixel, también llamados "punto a punto":

1. Modificación del histograma.
2. Modificación de la escala de niveles de gris.
- Métodos de vecindad:
 1. Suavización de la imagen.
 2. Mejora de contornos y detalles.
 3. Operaciones sobre las transformadas de la imagen.
 4. Engrosamiento.
 5. Eliminación de ruido.
 6. Pseudo color.

4.1 Histograma

El histograma de una imagen es una gráfica que representa la distribución de los valores de la intensidad de los píxeles, representando así las frecuencias relativas de los diferentes valores de los niveles de gris en la imagen. Un histograma típico es un gráfico bidimensional del número de ocurrencias del valor del pixel contra los valores de los píxeles, la figura 4.3 presenta el histograma de la figura 4.2, empleada como referencia en muchos de los artículos y libros sobre procesamiento de imágenes.

En un apartado más adelante se estudia la utilidad de obtener el histograma para el tratamiento de una imagen.

En la figura 4.3, el valor inferior corresponde al negro y el mayor al blanco.

El histograma de una imagen digital con niveles de gris en el rango $[0, L - 1]$ es una función discreta:

$$p(a_q) = \frac{n_q}{n}$$

donde:

- a_q : es el q -ésimo nivel de gris.
- n_q : es el número de píxeles en la imagen con nivel de gris a_q .
- n : es el número total de píxeles de la imagen.
- $q = 0, 1, 2, 3, \dots, L - 1$, siendo L el número de niveles de gris presentes en la imagen.
- $p(a_q)$ es un estimado de la probabilidad de ocurrencia de un nivel de gris a_q .



Figura 4.2: Imagen original de Lena empleada como referencia estándar en el procesamiento de imágenes [40].

La representación gráfica de esta función para todos los valores de q provee una descripción global de la apariencia de la imagen. El histograma da información acerca del brillo y el contraste de una imagen. Así, el histograma es una herramienta invaluable para el procesamiento de imágenes tanto cualitativa, como cuantitativamente.

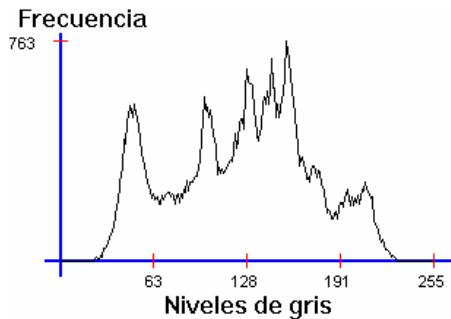


Figura 4.3: Histograma de la imagen de referencia. La abscisa representa los colores o niveles de gris en la imagen y la ordenada el número de píxeles que en la imagen tienen ese color.

4.2 Brillo y contraste

4.2.1 Brillo

El brillo es la intensidad o luminosidad de la imagen, es decir la tendencia a tonos claros u oscuros de cada pixel. Un cambio en el brillo de la imagen está caracterizado en el histograma por un corrimiento de la curva a lo largo del eje de las abscisas, como lo muestra la figura 4.4.

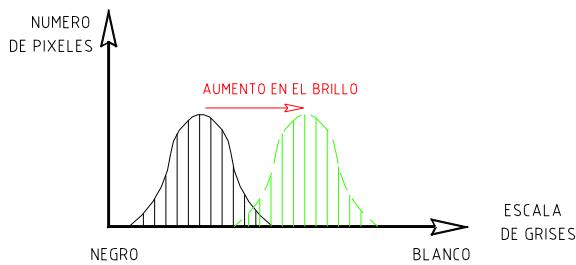


Figura 4.4: Cambio en el histograma al modificar el brillo en la imagen.

4.2.2 Contraste

El contraste indica la distribución existente en la imagen entre el máximo y el mínimo nivel de gris de los píxeles. Un cambio en el contraste de la imagen se puede visualizar en el histograma como una escalización de la curva únicamente en el sentido del eje de las abscisas. La figura 4.5 muestra este efecto.

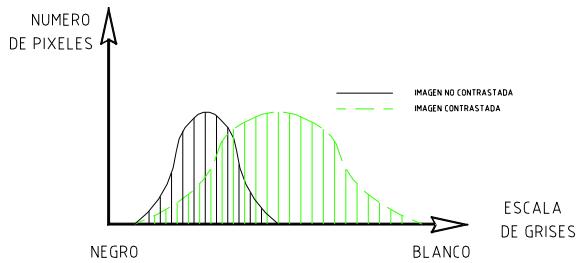


Figura 4.5: Cambio del histograma al modificar el contraste en una imagen.

4.3 Métodos basados en los pixeles (punto a punto)

A continuación se tratan las transformaciones simples que se aplican sobre cada pixel, independiente de sus vecinos y que permiten obtener mejoras en la calidad visual de una imagen.

4.3.1 Negativo de una imagen

En ciertas ocasiones, dadas las características no lineales de la respuesta del sistema visual humano, es conveniente emplear la imagen en negativo de una imagen dada. En la imagen en negativo pueden quedar más marcados ciertos detalles que pueden ser difíciles de percibir en la imagen original. Un claro ejemplo de su utilización se da en las radiografías. Para calcular la imagen en negativo o inversa se utiliza la siguiente formula:

$$q(x, y) = (L - 1) - p(x, y)$$

donde:

$p(x, y)$ corresponde al nivel de gris en la posición x, y en la imagen original,

$q(x, y)$ corresponde al nivel de gris resultante en la imagen en negativo,

L es el número máximo de niveles de gris que puede aparecer en una imagen.

En la figura 4.6 se presenta como ejemplo el negativo obtenido a partir de la imagen de referencia mostrada en la figura 4.2.



Figura 4.6: Negativo de la imagen de referencia.

4.3.2 Mejora del contraste

El contraste de una imagen puede verse reducido debido a diversos factores durante su adquisición, tales como : poca iluminación, sensores con un rango de valores exiguo, etc. Un tipo de mejora visual puede ser logrado al transformar los valores del nivel de gris de los pixeles de manera que se obtenga un aumento en el rango dinámico de estos valores. En general, se escogen dos niveles de gris, uno en la parte baja del rango correspondiente a los negros q_n y otro en la parte alta correspondiente a los blancos q_b , de tal manera que los pixeles con valor inferior o igual a q_n se transforman a un valor aun más bajo, tendiendo más hacia el negro y los valores superiores a q_b se convierten a un valor más alto, tendiendo al blanco, así:

$$\text{si } q_i = T(q_n) \Rightarrow q_i < q_n$$

$$\text{si } q_i = T(q_b) \Rightarrow q_i > q_b$$

y los pixeles con valores entre q_n y q_b se transforman siguiendo una transformación lineal que corresponda a los valores entre q_n y q_b . De esta manera se aumenta el rango dinámico de los valores centrales (entre q_n y q_b) del histograma. Esto se puede expresar matemáticamente de la siguiente forma:

$$q(x,y) = \begin{cases} a.p(x,y) & \text{para } p(x,y) < q_n, a < 1/2 \\ b.p(x,y) & \text{para } q_n < p(x,y) < q_b, b > 1/2 \\ c.p(x,y) & \text{para } p(x,y) > q_b, c < 1/2 \end{cases}$$

donde:

$p(x,y)$ corresponde al nivel de gris en la posición x,y en la imagen original,
 $q(x,y)$ corresponde al nivel de gris resultante en la imagen en negativo.

4.3.3 Compresión del rango dinámico

Existen ciertos casos en los cuales el rango dinámico del sensor de la imagen es mucho más amplio que aquel que puede ser representado en la imagen. En estos casos es necesario comprimir el rango dinámico para poder representar las imágenes dentro de los niveles de gris disponibles. La transformación más adecuada en estos casos es emplear una función monótona que permita comprimir los valores originales. Como ejemplo de esta función se puede presentar la raíz cuadrada o la función logarítmica.

Una expresión matemática de compresión es la siguiente:

$$s = c * \log(1 + |r|)$$

donde c es un factor de escala y la función \log realiza la compresión deseada.

4.3.4 Realce de rangos de intensidad

En ciertos casos se desea resaltar específicamente ciertos rangos de niveles de gris mientras que otros se ocultan. Esto resulta interesante cuando se quiere resaltar una característica específica de la imagen que se encuentra entre los niveles de gris resaltados.

4.4 Pretratamiento de imágenes

Las técnicas de pretratamiento de imágenes tienen por objetivo facilitar la segmentación aumentando la similitud entre los pixeles pertenecientes a una misma región y acentuando las diferencias entre los pixeles pertenecientes a distintas regiones. Entre estas técnicas podemos mencionar:

- Modificación del histograma.
- Reducción de ruido.
- Realce del contraste.

Dado el hecho de que las imágenes están compuestas por lo general de un gran número de pixeles, se busca aplicarles operadores locales que hacen uso de un número limitado de pixeles durante el cálculo.

4.5 Modificación del histograma

Las características de una imagen pueden ser modificadas a partir de la manipulación de su histograma. Se busca modificar la repartición de los niveles de gris de la imagen, de manera que el histograma de la imagen resultante corresponda al modelo de histograma fijado. Dado su carácter puntual, los métodos de transformación del histograma sólo modifican la apariencia visual de la imagen y no afectan la forma de las regiones.

Las técnicas descritas a continuación son técnicas potentes de mejora de imágenes: mejora del contraste, extensión de la dinámica, etc.

4.5.1 Condiciones para la aplicación de la modificación del histograma

Considerando primero el caso continuo y designando $a \geq 0$, la variable que representa los niveles de gris de la imagen a ser mejorada, considerada como una variable aleatoria de densidad de probabilidad $p_A(A[s])$. Se asume que los valores de los pixeles son cantidades continuas normalizadas en el intervalo $[0, 1]$, donde $a = 0$ representa el negro y $a = 1$ el blanco. Para cualquier a en el intervalo $[0, 1]$ se define la función:

$$b = T(a) \quad (1)$$

que produce un nivel b en la imagen resultante para cada valor de pixel a en la imagen original.

Se asume que:

La función de transformación $T(a)$ es una función de transformación de valor único y monótonamente creciente entre $0 \leq a \leq 1$ para conservar el orden de negro a blanco en la escala de grises. (2)

Además $0 \leq T(a) \leq 1$ para $0 \leq a \leq 1$, preservando así el rango de los niveles de gris empleado. (3)

La figura 4.7 muestra una función de transformación que satisface estas condiciones.

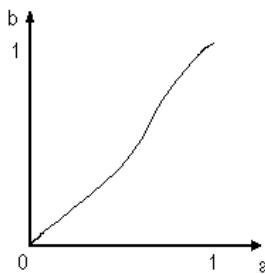


Figura 4.7: Relación entre una imagen y su transformada.

La transformación inversa es

$$a = T^{-1}(b) \text{ para } 0 \leq b \leq 1$$

donde se asume que T^{-1} también satisface las condiciones (2) y (3) con respecto a la variable b .

Los niveles de gris en la imagen pueden ser vistos como cantidades aleatorias en el intervalo $[0, 1]$. Si fueran variables continuas podrían ser caracterizadas por sus funciones de densidad de probabilidad $p_a(a)$ y $p_b(b)$.

De la teoría de probabilidad elemental, si $p_a(a)$ y $T(a)$ son conocidos y $T^{-1}(b)$ satisface la condición (2), se puede definir una función de densidad de probabilidad de la transformada de niveles de gris de la siguiente forma:

$$p_b(b) = \left[p_a(a) \frac{da}{db} \right]_{a=T^{-1}(b)} \quad (4.1)$$

Las técnicas de mejora de imágenes siguientes están basadas en la modificación de la apariencia de la imagen controlando la función de densidad de probabilidad de los niveles de gris por medio de la función de transformación $T(a)$.

4.5.2 Ecualización o linealización

Debido a que el ojo humano es más sensible al contraste que a la intensidad total de la imagen, se percibe menos información cuando la imagen tiene una pobre distribución de intensidades. La transformación de la imagen mediante la ecualización del histograma es un proceso que redistribuye sus intensidades con el fin de obtener un histograma lo más uniforme o plano posible.

La imagen está asociada a una modelización probabilística donde el histograma permite a menudo reforzar el contraste de los detalles de la imagen que son enmascarados por las variaciones de intensidad de mayor amplitud y escala.

Sea la transformación:

$$b = T(a) = \int_a^b p_a(w) dw \quad 0 \leq a \leq 1 \quad (4.2)$$

donde w es una variable de integración ficticia.

El lado derecho de la ecuación 4.2 corresponde a la función de distribución acumulativa de a . Las condiciones (2) y (3) se cumplen para $T(a)$.

De la ecuación 4.2 se puede encontrar:

$$\frac{db}{da} = p_a(a) \quad (4.3)$$

Reemplazando la ecuación 4.3 en la 4.1 se tiene:

$$\begin{aligned} p_b(b) &= \left[\frac{db}{da} \frac{da}{db} \right]_{a=T^{-1}(b)}, \text{ luego} \\ p_b(b) &= [1]_{a=T^{-1}(b)}, \text{ luego} \\ p_b(b) &= 1 \text{ para } 0 \leq b \leq 1 \text{ (Ver figura 4.8).} \end{aligned}$$

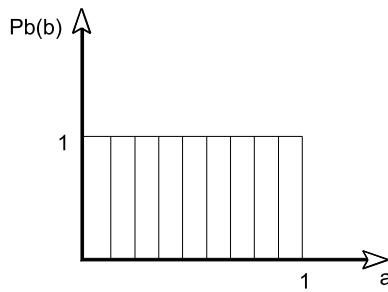


Figura 4.8: Función de densidad de probabilidad resultante de la imagen transformada.

La densidad de probabilidad de la variable transformada b en el intervalo de definición $[0, 1]$ es la ley uniforme. Se habla entonces de la uniformación de la densidad. Este resultado es independiente de la función de transformación inversa. Es decir, que cuando se emplee una función de transformación igual a la función de distribución acumulada de a se produce una imagen cuyos niveles de gris tienen una densidad uniforme.

En el caso discreto, se tiene:

$$p_a(a_i) = \frac{n_i}{n} \text{ para } 0 \leq a_i \leq 1 \text{ y } 0 \leq a_i \leq 1 \quad (4.4)$$

La forma discreta de la ecuación 4.2 viene dada por:

$$b_i = T(a_i) = \sum_{j=0}^i \frac{n_j}{n} \quad (4.5)$$

Reemplazando 4.4 en 4.5 se tiene:

$$b_i = T(a_i) = \sum_{j=0}^i p_a(a_j) \text{ para } 0 \leq a_i \leq 1 \text{ y } i = 0, 1, \dots, L-1$$

La transformación inversa está dada por:

$$a_i = T^{-1}(b_i) \text{ para } 0 \leq b_i \leq 1$$

donde $T(a_i)$ y $T^{-1}(b_i)$ deben verificar las condiciones (2) y (3).

La salida b del algoritmo de ecualización se escribe:

$$b'_i = Ent \left[\frac{b_i - b_{\min}}{1 - b_{\min}} (L - 1) + 0.5 \right]$$

donde:

- $\text{Ent}[x]$ representa la parte entera de x .
- b_{\min} es el más pequeño valor positivo de b .
- b' no será sino aproximadamente uniforme dado que b_k no lo es, pues es tan solo una aproximación discreta del caso continuo.

La figura 4.9 presenta la imagen resultante y su histograma al ecualizar la figura de referencia. Como puede observarse el histograma no es absolutamente uniforme, pero está mucho mejor distribuido. En la práctica, es muy difícil obtener un histograma absolutamente uniforme debido a que la técnica resulta de una aproximación discreta del caso continuo. Esta transformación incrementa el rango dinámico de los niveles de gris en la imagen por lo cual se produce un incremento en el contraste, el cual mejora en general la apariencia visual de la imagen [20]. Sin embargo las imágenes, cuyo histograma es angosto contenido pocos niveles de gris, pueden presentar como consecuencia negativa un incremento en la percepción granular de la imagen, la impresión de estar hecha por parches [20] y resaltar algunas irregularidades debidas al muestreo.

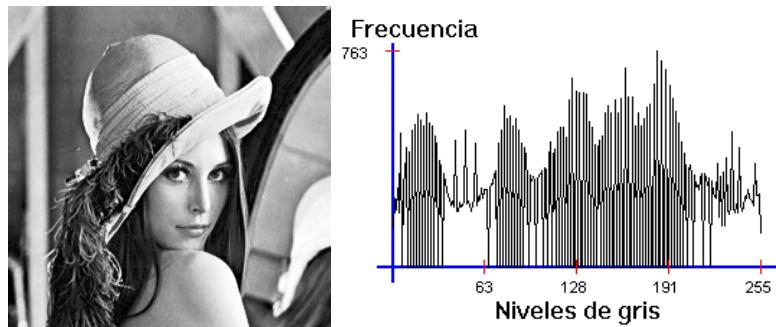


Figura 4.9: Imagen ecualizada y su histograma.

4.5.3 Especificación del histograma

Teniendo una imagen de partida a y definiendo c como la imagen que se quiere obtener, entonces se puede establecer:

- Una imagen b resultante de una transformación de la imagen a :

$$b = T(a) = \int_0^a p_a(w) dw$$

- Una imagen d resultante de una transformación de la imagen c :

$$d = Q(c) = \int_0^c p_c(w) dw$$

por tanto:

$$c = Q^{-1}(d) = Q^{-1}(b) = Q^{-1}[T(a)]$$

4.6 Transformación local del histograma

Las técnicas de modificación del histograma antes vistas se aplican sobre la totalidad de la imagen, sin embargo una aplicación local puede ser realizada con el fin de resaltar más los cambios en pequeñas zonas de la imagen. Así, la imagen puede ser dividida en zonas generalmente cuadradas y la técnica puede ser aplicada sobre el histograma calculado sobre cada una de las zonas. Sin embargo, esta técnica a veces produce como resultado un efecto como de tablero de ajedrez dada la aplicación excluyente entre una zona y otra. Por tanto, se puede hacer una variación sucesiva de regiones dadas de la imagen, delimitadas por una ventana móvil cuadrada o máscara que se desplaza con un sentido específico sobre toda imagen. La figura 4.10 muestra la secuencia de barrido de la ventana móvil.

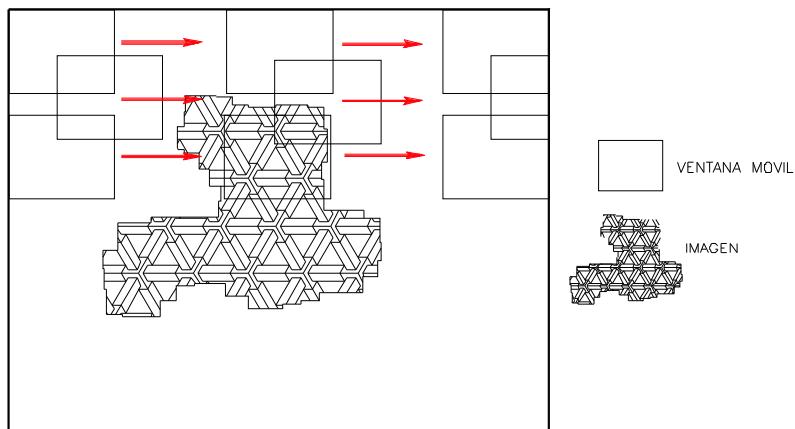


Figura 4.10: Transformación local de una imagen.

Además del histograma, la mejora también puede basarse en otras propiedades de los píxeles. El valor promedio y la varianza o la desviación estándar son otras variables que pueden ser usadas, dada su importancia en el aspecto de la imagen. Así, la media es una medida del promedio del brillo y la varianza es una medida del contraste [20]. Una transformación local basada en estos criterios se puede expresar en la siguiente ecuación:

$$q(x, y) = f(x, y) [p(x, y) - m(x, y)] + m(x, y) \quad (4.6)$$

donde:

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

- $p(x, y)$ es el nivel de gris de la imagen original.
- $q(x, y)$ corresponde al nivel de gris de la imagen resultante.
- $m(x, y)$ es la media de los niveles de gris calculados en el entorno de (x, y) .
- $\sigma(x, y)$ es la desviación estándar de los niveles de gris calculados en el mismo entorno.
- M es la media global calculada sobre toda la imagen de $a(x, y)$.
- k es una constante definida en el intervalo dado.

Como se observa, los valores de f , m y σ dependen del tamaño de ventana escogido alrededor de (x, y) . f se comporta como un valor de ganancia que amplifica las variaciones entre el pixel a y la media local m , amplificando de esta manera las variaciones locales. Dado que f es inversamente proporcional a la desviación estándar de los niveles de gris, las zonas con muy poco contraste tendrán una amplificación mayor.

En la ecuación 4.6 se ha adicionado la media m con el fin de mantener el valor promedio de intensidad en la zona. En la práctica a veces es conveniente la adición posterior de una porción de la media local y limitar las variaciones de q entre dos valores $q_{\min} < q < q_{\max}$ para evitar grandes variaciones de la intensidad en regiones aisladas.

4.7 Imagen promedio

Una técnica bastante utilizada con el fin de mejorar la calidad de una imagen consiste en realizar un promedio de varias imágenes obtenidas del mismo objeto. Este procedimiento supone que cada imagen real obtenida $q(x,y)$ está constituida de la imagen ideal $p(x,y)$ más un ruido aditivo $n(x,y)$, el cual es una función aleatoria de media nula y no correlacionada entre cada uno de los puntos (x,y) de la imagen. Por consiguiente, la i -ésima imagen adquirida puede ser representada como:

$$q_i(x,y) = p(x,y) + n_i(x,y)$$

Promediando M imágenes reales $q_i(x,y)$ se obtiene:

$$\bar{q}(x,y) = \frac{1}{M} \sum_{i=1}^M q_i(x,y) = p(x,y) + \frac{1}{M} \sum_{i=1}^M n_i(x,y)$$

Dado que el ruido $n(x,y)$ tiene media nula, la esperanza matemática de $\bar{q}(x,y)$ es:

$$E[\bar{q}(x,y)] = p(x,y)$$

$\bar{q}(x,y)$ es un estimador sin sesgo de la imagen ideal $p(x,y)$.

Si la varianza de la función de ruido $n(x,y)$ es igual a $\sigma_n^2(x,y)$, la varianza del estimador $\bar{q}(x,y)$ es:

$$V[\bar{q}(x,y)] = \sigma_{\bar{q}}^2(x,y) = E[\bar{q}(x,y) - E[\bar{q}(x,y)]]^2$$

$$V[\bar{q}(x,y)] = E\left[\frac{1}{M} \sum_{i=1}^M n_i(x,y)\right]^2 = \frac{1}{M} \sigma_n^2(x,y)$$

de donde se determina que la desviación estándar está dada por:

$$\sigma_{\bar{q}}(x,y) = \frac{1}{\sqrt{M}} \sigma_n(x,y)$$

En consecuencia, promediando M imágenes reales $q_i(x,y)$, la relación señal a ruido es mejorada por un factor de \sqrt{M} .

4.8 Substracción de imágenes

Una técnica empleada para resaltar ciertas características que pueden diferenciar una imagen de otras (entre una toma y otras del mismo objeto) y que suele ser empleada en medicina y astronomía, consiste en hacer una resta entre éstas. Esta técnica puede ser útil cuando el proceso de adquisición de las imágenes no garantiza que éstas se tomen con la misma intensidad de luz, como es el caso en imágenes astronómicas y satelitales.

En medicina, esta técnica permite, por ejemplo, detectar las zonas por donde pasa o es absorbida una determinada sustancia ingerida o inyectada en un paciente.

Este procedimiento puede corregir igualmente ruidos y artefactos que pueden aparecer entre una toma y otra como consecuencia de errores o desperfectos en el aparato de medida [42].

La definición formal del método puede expresarse en la siguiente forma:

Dadas dos imágenes definidas por las funciones $f(x,y)$ y $p(x,y)$ se puede lograr una tercera imagen $q(x,y)$, igual a la diferencia de las imágenes f y p :

$$q(x,y) = f(x,y) - p(x,y)$$

4.9 Filtrado espacial

Este tipo de filtraje considera todas las operaciones directas sobre un pixel y todos sus vecinos y su principal atributo: el nivel de gris. El concepto de este tipo de filtros está asociado directamente con el de los filtros en el dominio de la frecuencia (ver figura 5.14). Así, estos filtros tienen igualmente el propósito de modificar la contribución de ciertos rangos de frecuencia que aparecen en la imagen original. Por lo tanto, si el objetivo buscado es el de atenuar el ruido y hacer desaparecer los defectos y las perturbaciones contenidas en la imagen inicial; se pueden realizar operaciones de suavización obteniendo resultados análogos a los obtenidos por medio de un filtro pasa-bajos frecuencial o bien si se desea realzar ó acentuar los bordes, se puede utilizar un filtro cuyo resultado se asemeja al obtenido luego de aplicar un filtro pasa-altos frecuencial. Además, existe la posibilidad de implementar filtros pasabandas para realzar cualquier patrón de comportamiento que esté asociado a un determinado rango de frecuencias en la formación de la imagen. Así, estos filtros son empleados básicamente en la reconstrucción de imágenes, pero son de poco interés en la mejora de éstas.

Aunque la teoría del filtrado lineal se ha desarrollado en general en el dominio de la frecuencia, el teorema de la convolución permite trasladarla al dominio espacial, aunque su comprensión ya no es tan intuitiva. Sea $Q(u,v)$ la transformada de Fourier de la imagen original y $H(u,v)$ la función de transferencia del filtro a utilizar, se tiene que:

$$F^{-1}[Q(u,v) \cdot H(u,v)] = q(x,y) * h(x,y)$$

donde el símbolo $*$ representa el operador de convolución.

De aquí se puede determinar la forma que adoptan las funciones de transferencia $h(x,y)$, conocidas como filtros espaciales, correspondientes a cada uno de los filtros frecuenciales. La figura 4.11 presenta las formas de los filtros frecuenciales básicos y sus correspondientes en el dominio espacial.

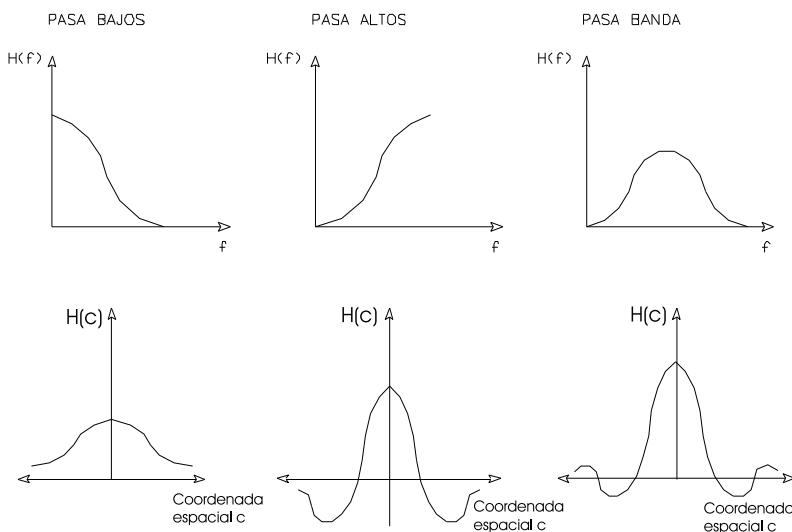


Figura 4.11: Curvas características en la frecuencia y en el espacio de los filtros pasa-bajos, pasa-altos y pasa-banda.

Teóricamente, el filtro espacial $h(x,y)$ debería calcularse sobre una malla o ventana de tamaño igual al de la imagen $q(x,y)$ sobre la cual se aplica el filtro. Dado el hecho de que los valores de $h(x,y)$ decrecen rápidamente alrededor del origen, se acostumbra realizar este tipo de filtrado mediante la utilización de ventanas, conocidas como máscaras de convolución, que son mucho más pequeñas y aproximan la forma de $h(x,y)$. Cada una de éstas puede ser vista como una ventana que se aplica sobre cada pixel de la imagen, uno por uno, en la siguiente forma (ver figura 4.12):

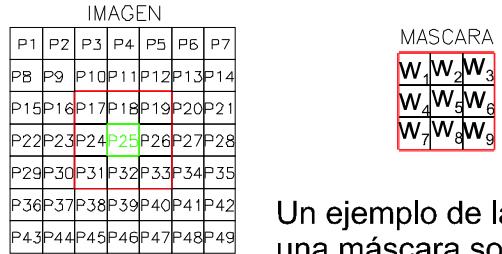
Suponiendo una máscara, la cual tiene un valor nulo fuera de su propia ventana de dimensión 3x3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

aplicada sobre un grupo de pixeles $q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8$ y q_9 , y centrada sobre q_5 ; el valor del pixel en la imagen resultante q'_5 es calculado empleando los coeficientes de la máscara en una suma ponderada de los valores del pixel q_5 y sus vecinos:

$$q'_5 = w_1 \cdot q_1 + w_2 \cdot q_2 + w_3 \cdot q_3 + w_4 \cdot q_4 + w_5 \cdot q_5 + w_6 \cdot q_6 + w_7 \cdot q_7 + w_8 \cdot q_8 + w_9 \cdot q_9 \quad (4.7)$$

siendo las máscaras más utilizadas de tamaño 3x3, 5x5 y 7x7.



Un ejemplo de la aplicación de una máscara sobre el pixel p_{25} de la imagen es:

$$\begin{aligned} p'_{25} = & p_{17} \cdot xw_1 + p_{18} \cdot xw_2 + p_{19} \cdot xw_3 + p_{24} \cdot xw_4 \\ & + p_{25} \cdot xw_5 + p_{26} \cdot xw_6 + p_{31} \cdot xw_7 + p_{32} \cdot xw_8 \\ & + p_{33} \cdot xw_9 \end{aligned}$$

Nota: Este proceso se repite sobre todos los pixeles, excepto sobre los bordes, en el orden mostrado.

Figura 4.12: Aplicación de una máscara de convolución sobre una imagen.

Vale la pena resaltar que el proceso de filtrado usando la transformada de Fourier puede resultar más rápido que el usar máscaras de convolución cuando la máscara es de gran tamaño (alrededor de 13x13 o más).

4.9.1 Filtro pasabajos

Este filtro reduce los efectos de ruido introducidos en la imagen, ya sea en el proceso de adquisición o de digitalización. El corta los valores extremos o picos de los niveles de gris en la imagen a partir de un valor dado. La imagen resultante se caracteriza por la reducción de la cantidad de detalles finos y la eliminación de texturas irrelevantes, dando la impresión de hacer la imagen borrosa.

El filtro es básicamente una extensión bidimensional del filtro pasabajos unidimensional de ventana móvil empleado en procesamiento de señales. Consiste en reemplazar cada pixel de la imagen por el valor promedio de los niveles de gris del pixel y de sus vecinos, operación análoga a la integración. Así:

$$b(x, y) = \frac{1}{n} \sum_{(i,j) \in M} q(i, j)$$

donde M es el conjunto de n -vecinos del pixel ubicado en (x, y) sobre el cual se aplica el filtro, n es el número de pixeles de M y $q(i, j)$ es el nivel de gris de cada pixel.

Este promedio se obtiene generalmente mediante la utilización de una máscara de convolución simétrica de talla 3x3, 5x5 ó 7x7 aplicada alrededor de cada pixel, tal como:

$$\frac{1}{9} x \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

En la figura 4.13 se presenta el resultado de aplicar sobre la imagen de referencia una máscara pasa-bajos de dimensión 3x3.



Figura 4.13: Imagen resultante al aplicarle el filtro pasa-bajos.

Como se observa, la imagen aparece un poco borrosa y da la impresión de estar desenfocada, por este motivo esta clase de máscaras son conocidas como máscaras de suavización o desenfoque. Para reforzar el efecto de desenfoque se puede emplear una máscara más grande o repetir el proceso de filtraje varias veces, en este caso la imagen resultante $q(x, y)$ está definida por:

$$q(x, y) = p(x, y) * h(x, y) * h(x, y) * \dots * h(x, y)$$

El filtro pasabajos o de suavización, se puede generalizar de la siguiente manera:

1. Máscara de dimensión $n \times n$, donde n es impar.
2. Los pesos de la máscara ω son simétricos alrededor del pixel central.
3. Todos los pesos $\omega_i \geq 0$, donde el peso central es mayor a cero.
4. Los pesos ω_i se mantienen constantes o disminuyen en su valor a medida que se alejan del centro.

Además, $\sum \omega_i = 1$, para no aumentar o disminuir el brillo de la imagen. El efecto de suavización aumenta al incrementar el valor de n . El perfil de los pesos en la máscara: exponencial, gaussiana, piramidal, parabólica, rectangular, etc. da diferencias cualitativas en la imagen final.

La figura 4.14 presenta el resultado luego de aplicar a la imagen de referencia el siguiente filtro:

0.05	0.15	0.05
0.15	0.2	0.15
0.05	0.15	0.05

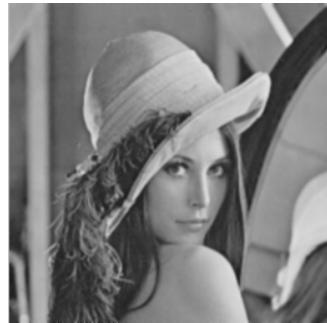


Figura 4.14: Imagen suavizada.

Besançon [3] presenta un operador pasabajos utilizado en el programa comercial VICOM y caracterizado por la máscara normalizada:

$$h_{Pasabajos}(x, y) = \frac{1}{a+2} \begin{bmatrix} 1 & a & 1 \\ a & a^2 & a \\ 1 & a & 1 \end{bmatrix}$$

$$0 \leq a \leq 9$$

Se puede notar que en el caso $b = 1$, se obtiene la máscara típica del filtro pasabajos visto antes. Cuando $a = 0$ el filtro no sigue la forma de la curva característica del pasabajos vista en la figura 4.11, ni cumple las especificaciones definidas en la generalización de una máscara pasabajos vistas antes. Sin embargo, presenta un comportamiento de filtro pasabajos debido al promedio que realiza sobre la señal alrededor del pixel central.

4.9.2 Filtro gaussiano

Otra máscara de filtraje de ruido es obtenida a partir de la función de Gauss:

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

donde σ es la desviación estándar que representa el ancho de la distribución Gaussiana mostrada en la figura 4.15.

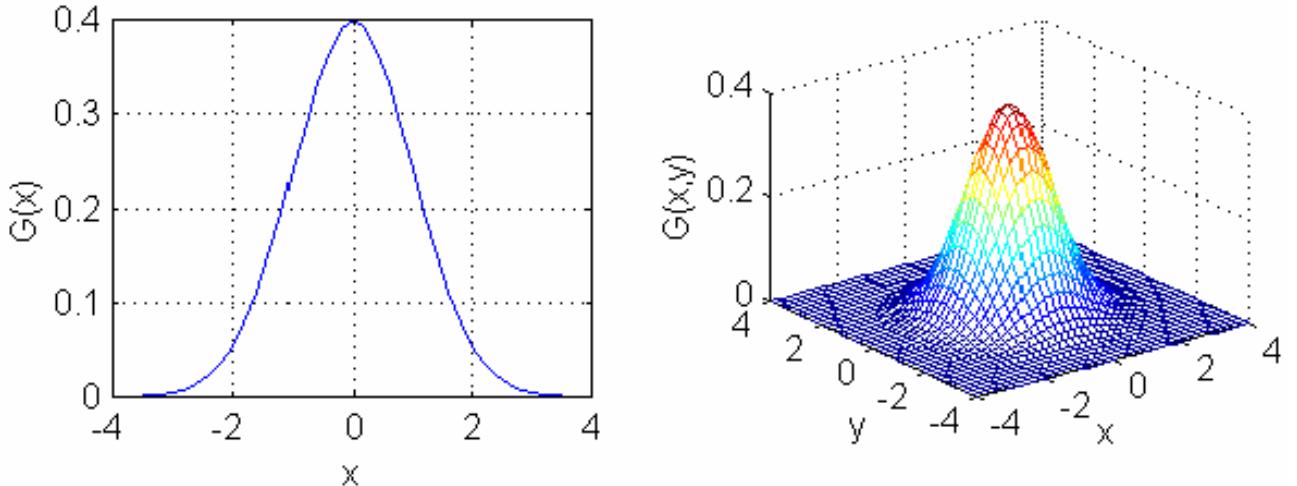


Figura 4.15: Distribución gaussiana de una y dos variables para $\sigma = 1$.

En el caso bidimensional, la imagen es entonces suavizada aplicando un filtro dado por:

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

como el que se ilustra en la figura 4.15.

La forma de la distribución y por consiguiente la magnitud de la suavización puede ser controlada variando el valor de σ , que permite definir que tan rápido cae la curva a partir de su valor central.

Para filtrar una imagen $p(x, y)$, se convoluciona con $G(x, y)$ para producir una imagen filtrada, es decir, $q(x, y) = p(x, y) * G(x, y)$. La máscara de convolución gaussiana es una máscara circularmente simétrica (isotrópica), hecha de tal manera que a cada uno de sus pesos se le asigna el valor de la función gaussiana en el punto.

Para definir los pesos se calcula una máscara de radio r (distancia en pixeles del centro de la máscara) de acuerdo al valor de σ . Dado que la función gaussiana no cae a cero para cualquier valor finito de r , es necesario truncar la función cuando los pesos son suficientemente pequeños para ser considerados insignificantes, con el fin de mantener la máscara de una talla razonablemente pequeña. A pesar de que una máscara discreta es sólo una aproximación de la función Gaussiana, puesto que ha sido muestreada y truncada, es suficientemente buena

para su uso práctico. Para evitar modificar el nivel de gris promedio de la imagen transformada se normalizan los pesos de la máscara. σ puede ser visto como un parámetro de ajuste de la talla de las estructuras a eliminar; de esta manera, si las estructuras tienen un tamaño menor comparado con el valor de σ , son removidas, de lo contrario permanecen en la imagen.

En la figura 4.16 se muestra la imagen resultante al aplicar el filtro de Gauss sobre la imagen de Lena con $\sigma = 1$ y una máscara de tamaño 5×5 .



Figura 4.16: Imagen tratada con filtro Gaussiano con $\sigma = 1$ y usando una máscara de tamaño 5×5 .

La convolución es fácil de implementar, tal como se mostró antes. Pero presenta la desventaja de que su costo computacional es bastante alto. Sin embargo, el filtro gaussiano tiene la propiedad de ser separable, es decir, puede ser realizado mediante dos convoluciones en direcciones ortogonales con máscaras gaussianas unidimensionales, lo cual puede ser demostrado a partir de la definición de la función [49].

El filtro gaussiano tiene varias propiedades de optimización: remueve el ruido y las texturas de menor tamaño dado que estas contienen una alta variación de niveles de gris correspondientes a altas frecuencias. El objeto de la suavización es el de remover las altas frecuencias sin distorsionar las frecuencias bajas, lo cual es posible realizar con la función de Gauss gracias a su característica de ser en sí misma una curva suavizada; por lo tanto, es particularmente buena para separar las altas y bajas frecuencias usando sólo la información necesaria de una más amplia zona de la imagen.

Muchos sistemas de procesamiento de imágenes se sirven de las diversas estructuras en diferentes escalas presentes en la imagen, guardando múltiples representaciones obtenidas luego de filtrar la imagen original con máscaras de Gauss con diferentes valores de σ . Cuando una imagen ha sido suavizada, no es necesario conservar la misma cantidad de píxeles que la imagen original, por ejemplo, luego de suavizar la imagen con un filtro de Gauss con $\sigma = 2$, tan solo la cuarta parte de los píxeles es necesaria para representar la imagen. Así, las imágenes suavizadas pueden ser conservadas en arreglos de menor tamaño, resultando en una estructura de datos llamada de resolución piramidal [49].

4.9.3 Filtros de realce

Al aplicar filtros de realce se busca resaltar aquellas características de la imagen que pueden haber resultado debilitadas en el proceso de adquisición de la imagen, reforzando los bordes o contornos de los objetos presentes en la imagen. Ellos realzan los detalles finos, pero intensifican el ruido.

4.9.4 Filtro pasaaltos

El filtro pasa altos se suele emplear para resaltar los contornos de la imagen. Tal como se observa en la figura 5.14, la característica espacial de este filtro tiene un valor central positivo, los valores laterales son negativos y fuera de la máscara los valores son iguales a cero. Así se tiene una máscara:

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

Una característica importante de este tipo de máscaras es que la suma de sus coeficientes es igual a cero, de manera que al aplicarla sobre una zona donde la imagen es constante o no tenga grandes variaciones el

resultado del filtrado será cero o un valor pequeño. Como se puede observar la máscara de convolución es aproximadamente isotrópica, así esta puede ser rotada y el efecto es siempre el mismo, en esta forma ella funciona aproximadamente igual en todas las direcciones. Si el operador es aplicado sobre una región uniforme de la imagen, su salida será cero, dado que los ocho pesos de -1 se cancelan con el de 8 del pixel central. Si por el contrario, el operador se aplica sobre un borde, el resultado será positivo o negativo, dependiendo del lado donde se halla aplicado la máscara.

En la figura 4.17 se puede observar el efecto de "borde" de este filtro. Esta máscara como se ve luego, implementa una aproximación matemática del operador Laplaciano, el cual a su vez es cercanamente aproximado a la derivada segunda de una función. Este filtro se caracteriza por hacer desaparecer las bajas frecuencias y como consecuencia de ésto el contraste de la imagen disminuye fuertemente.



Figura 4.17: Imagen tratada con filtro pasa-altos.

4.9.5 Filtro de realce

Una definición alternativa para el filtro pasaaltos es la siguiente:

$$\text{Imagen}_{Pasaaltos} = \text{Imagen}_{Original} - \text{Imagen}_{Pasabajos} \quad (4.8)$$

A partir de esta ecuación, se puede definir un filtro de realce como:

$$\text{Imagen}_{Realizada} = k \cdot \text{Imagen}_{Original} - \text{Imagen}_{Pasabajos} \quad k > 1 \quad (4.9)$$

Donde k es la constante de realce.

A partir de la ecuación 4.8 y reemplazando en la ecuación 4.9 se obtiene:

$$\text{Imagen}_{Realizada} = (k - 1) \cdot \text{Imagen}_{Original} + \text{Imagen}_{Pasaaltos}$$

A partir de esta ecuación, se concluye que si $k=1$ se obtiene la imagen filtrada con el filtro pasa-altos y si $k>1$ parte de la imagen original se añade al resultado de la imagen filtrada con el filtro pasa-altos, por lo que se obtendrá la imagen original con los bordes realzados, tal como se ve en la figura 4.18.

Una forma de implementar este filtro en una máscara es:

$$\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 * k - 1 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad k \geq 1$$



Figura 4.18: Imagen tratada con filtro de realce, $k = 1$.

Se pueden definir las características de un filtro de realce de la siguiente manera [36]:

1. Máscara de tamaño $n \times n$, donde n es impar.
2. Los pesos de la máscara ω_i son simétricos alrededor del pixel central.
3. La suma de los pesos centrales es mayor que 0.
4. Los pesos que rodean a los pesos centrales son negativos y fuera de la máscara son cero.
5. $\sum \omega_i > 0$.
6. Se debe normalizar para no aumentar el contraste en la imagen resultante.

La figura 4.19 presenta el resultado, luego de aplicar a la imagen de referencia la máscara siguiente:

0	-1	0
-1	5	-1
0	-1	0



Figura 4.19: Imagen tratada con filtro de realce.

El efecto de realce aumenta a medida que n disminuye y reduciendo el número de pesos positivos centrales.

Además de los filtros cuyo comportamiento es lineal, de acuerdo a la ecuación 4.7, hay varios tipos de filtros que tienen una característica no lineal, entre estos podemos citar:

- Filtro del valor medio.
- Filtro del máximo.
- Filtro del mínimo.
- Filtro mediana.

4.9.6 Filtro del máximo

Este filtro compara los valores de nivel de gris de los pixeles vecinos a un pixel determinado y asigna a éste el valor máximo de los pixeles comparados:

$$q'_5 = \max\{q_k \mid k = 1, \dots, n\}$$

donde n es el número de píxeles incluidos en la máscara.

4.9.7 Filtro del mínimo

Este filtro compara los valores de nivel de gris de los píxeles vecinos a un pixel determinado y le asigna el valor mínimo de los píxeles comparados:

$$q'_5 = \min\{q_k \mid k = 1, \dots, n\}$$

donde n es el número de píxeles incluidos en la máscara.

4.9.8 Filtro mediana

Un método bastante efectivo para la eliminación de ruido sin volver difusa la imagen consiste en la utilización del filtro mediana. Se define la mediana m de un conjunto de $2n+1$ valores como el valor medio de este conjunto de números. La mediana de una distribución no es dañada por un valor aberrante. En el caso de una imagen, se escoge una ventana de lado $n \times n$, n impar, centrada sobre el punto a filtrar y se aplica un simple método de ordenamiento de los píxeles seleccionados en la máscara. Este filtro selecciona el valor medio de los píxeles limitados en la máscara y asigna este valor al pixel central.

Este método resulta especialmente útil en la eliminación de ruido de tipo impulsivo. Además no distorsiona la imagen, pero en contrapartida, produce una pequeña pérdida de resolución.

4.9.9 Filtro de realce en relieve

El realce en relieve hace que una imagen aparezca como si estuviera hecha en piedra, siendo usado como un efecto visual fotográfico [36]. Para ello la máscara debe cumplir las siguientes características:

1. Máscara de dimensión $n \times n$ donde n es impar.
2. Los pesos de la máscara ω_i deben ser valores asimétricos (positivos y negativos) alrededor del pixel central y cancelándose uno a otro.
3. $\sum \omega_i = 0$.
4. La suma de los pesos centrales es igual a 0.

Un primer ejemplo se muestra en la figura 4.20, donde se empleó la siguiente máscara:

-1	0	0
0	0	0
0	0	1



Figura 4.20: Figura tratada con filtro de realce en relieve.

Un segundo ejemplo se muestra en la figura 4.21, donde se empleó la siguiente máscara:

1	4	-2
3	0	-3
2	-4	-1



Figura 4.21: Figura tratada con filtro de realce en relieve.

4.9.10 Detección de bordes

La detección de bordes se basa en la ubicación de discontinuidades locales que corresponden principalmente a las fronteras de los objetos en la imagen. Un borde, también conocido como una discontinuidad en una señal, se define comúnmente como una transición o cambio significativo en un parámetro físico de la imagen tal como su intensidad o amplitud, su color o su textura.

Típicamente, un borde ideal en una dimensión puede modelarse como una transición desde una intensidad baja a una alta. En la práctica, la señal se encuentra degenerada comúnmente por el ruido, tal como se muestra en la figura 4.22.

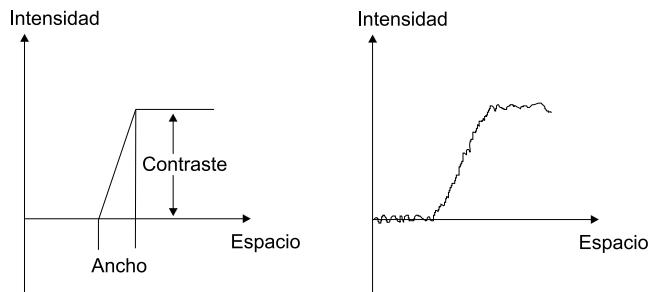


Figura 4.22: Modelo unidimensional de un borde rampa.

Tales bordes están caracterizados por su contraste (diferencia entre las intensidades alta y baja) y su ancho o rata de cambio. Estas medidas diferencian los bordes bruscos de los suaves.

En dos dimensiones, las discontinuidades ocurren a lo largo de ciertas líneas u orientaciones, tal como se ilustra en la figura 4.23. Al igual que en el caso unidimensional, un ancho o rata de cambio y un contraste pueden ser definidos. Además, el aspecto de la dirección u orientación local es importante para caracterizar los bordes en imágenes bidimensionales.

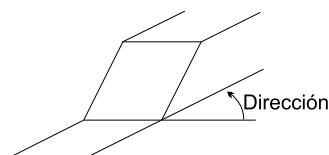


Figura 4.23: Modelo bidimensional de un borde rampa.

Este modelo, en forma de rampa, no es único y existen como se mencionó antes, otras definiciones de bordes usando el perfil de transición o las propiedades de textura (ver figura 4.24) y color de los objetos en la imagen [1].

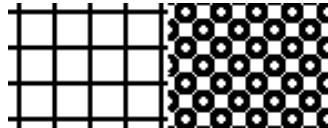


Figura 4.24: Modelo de borde entre dos texturas.

Algunos ejemplos de otras formas de bordes se muestran en la figura 4.25 y varios estudios han conducido a diversos detectores para tales bordes. Si se analiza el cambio en el valor de la intensidad en una dirección, se pueden encontrar entre otras las siguientes clases de bordes:

escalón, paso, techo convexo, techo cóncavo, etc.

La discontinuidad de tipo escalón se presenta cuando los valores de los píxeles a través del borde cambian bruscamente.

Los bordes en forma de rampa tienen el mismo signo en los componentes del gradiente a ambos lados de la discontinuidad, mientras los bordes de techo tienen signos opuestos.

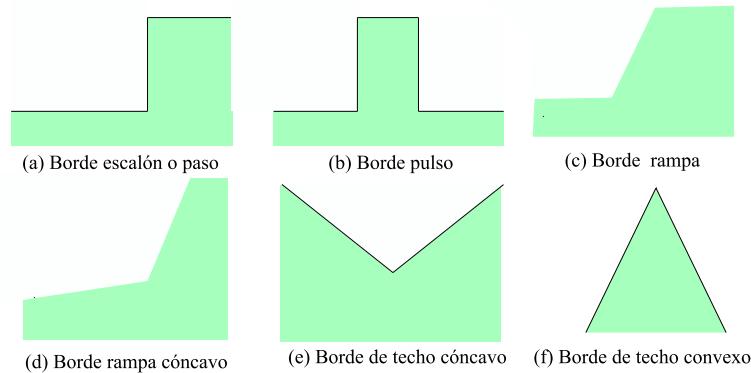


Figura 4.25: Algunos modelos de bordes en las discontinuidades de una imagen.

La discontinuidad paso es aquella que se presenta cuando los valores de los píxeles se incrementan rápidamente para luego disminuir nuevamente (o viceversa) a través de alguna línea. Este tipo de discontinuidad se puede presentar, por ejemplo, en el caso de una línea delgada negra colocada sobre un fondo blanco, tal como se muestra en la figura 4.26. Las discontinuidades paso ocurren cuando un objeto oculta a otro u otra parte de si mismo. Ellas pueden separar objetos diferentes u ocurrir cuando una sombra cae sobre de un objeto.

En el caso unidimensional (figura 4.26), el punto de transición puede marcarse como el máximo de la derivada de primer orden o el pasaje por cero de la derivada de segundo orden.

Note que la derivada de primer orden es más alta para un perfil brusco de transición que para una transición suave como se muestra en la figura 4.27.

Luego, en el caso 2D un pixel correspondiente al borde de un objeto puede verse a través de una discontinuidad o cambio brusco en el gradiente. Lo anterior sugiere dos etapas para la determinación de bordes:

Primero, resaltar los bordes locales usando técnicas de diferenciación. La extensión de tales derivadas de 1° y 2° orden al caso bidimensional se basan en el uso del gradiente y el laplaciano respectivamente. La primera realiza la detección basada en los cambios bruscos del gradiente y la segunda en el pasaje por cero del laplaciano.

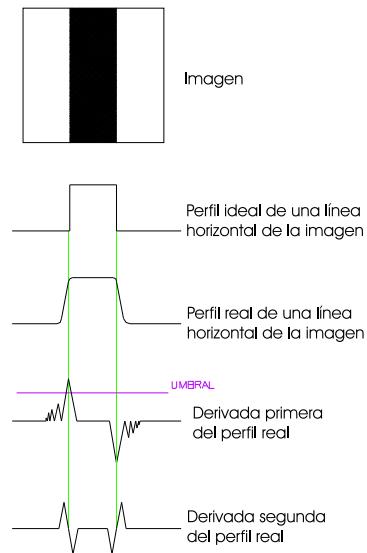


Figura 4.26: Señal, gradiente y laplaciano de una línea de la imagen.

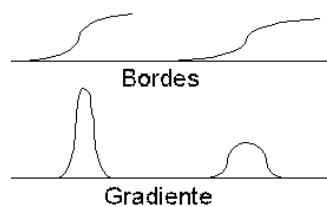


Figura 4.27: Gradiente correspondiente a dos tipos de borde: brusco y suave.

Segundo, evaluar el resultado para conservar únicamente como puntos de borde los puntos con derivadas suficientemente grandes, mayores que un valor de umbral. El valor de umbral es muy importante para la detección de los bordes, si este es muy grande no permite la detección de objetos en la imagen donde el gradiente no tenga demasiada amplitud y si es muy pequeño se puede detectar el ruido como un falso borde en la imagen.

4.9.11 Máscaras de gradiente

Contrario al efecto de integración del filtro pasa-bajos, la derivación resalta los bordes de la imagen. El tipo de filtro de derivación más empleado está basado en el cálculo del gradiente. El gradiente es un vector, cuyos componentes miden qué tan rápido cambian con la distancia los valores de los píxeles a lo largo de dos direcciones ortogonales. En general este se define en las direcciones x y y .

En el caso bidimensional continuo, el gradiente de una función $q(x,y)$ está definido de la siguiente forma:

$$\nabla_q(x, y) = \begin{bmatrix} \nabla_x \\ \nabla_y \end{bmatrix} = \begin{bmatrix} \frac{\partial q}{\partial x} \\ \frac{\partial q}{\partial y} \end{bmatrix} \quad (4.10)$$

donde ∇_x y ∇_y son los componentes del gradiente de la función f con respecto a x y con respecto a y respectivamente. Si imagina una superficie ubicada sobre una imagen, cuya altura en cada punto sea proporcional al nivel de gris en éste, es claro que una medida de la pendiente cuando se desplaza a lo largo de las direcciones x y y está dada por el gradiente.

La magnitud del gradiente está definida por:

$$|\nabla q(x, y)| = \sqrt{\nabla_x^2 + \nabla_y^2}$$

La magnitud del gradiente sirve de base para varias aproximaciones de filtros de diferenciación mediante el empleo de máscaras de convolución para su uso en imágenes.

Una aproximación bastante empleada para el cálculo de la magnitud del gradiente es la siguiente:

$$|\nabla q(x, y)| \approx |\nabla_x| + |\nabla_y|$$

ésta es más fácil de implementar en un programa y es más eficiente computacionalmente que la fórmula original.

La dirección del gradiente está dada por:

$$\theta(x, y) = \arctan \left(\frac{\nabla_y}{\nabla_x} \right)$$

donde el ángulo θ se mide con respecto al eje x .

Este operador puede ser aproximado en el caso discreto por operadores de diferencia implementados con máscaras de convolución. A continuación, se analiza el diseño de las máscaras y su aplicación sobre un pixel de una imagen, basado en la siguiente nomenclatura para una serie de píxeles en una imagen:

q_1	q_2	q_3
q_4	q_5	q_6
q_7	q_8	q_9

La ecuación 4.10 aplicada alrededor del punto q_5 puede aproximarse de distintas formas, pero debe anotarse que la suma de todos los coeficientes de una máscara de gradiente es siempre igual a cero, produciendo una respuesta igual a cero en regiones constantes, donde no hay cambios, tal como se espera de un operador diferencial.

A. Mascara de Robert

En imágenes discretas las derivadas parciales $\frac{\partial q}{\partial x}$ y $\frac{\partial q}{\partial y}$ se pueden aproximar aplicando un operador de diferencia:

$$\nabla_x = \frac{\partial q(x, y)}{\partial x} \approx \Delta x = \frac{q(x + d_x, y) - q(x, y)}{d_x}$$

$$\nabla_y = \frac{\partial q(x, y)}{\partial y} \approx \Delta y = \frac{q(x, y + d_y) - q(x, y)}{d_y}$$

donde d_x y d_y son intervalos a lo largo de las direcciones x y y respectivamente.

En imágenes discretas d_x y d_y pueden ser consideradas por el número de píxeles entre dos puntos. Así, si $d_x = d_y = 1$ (espaciado igual a un pixel) y estando en un punto de coordenadas (x, y) se tiene:

$$\Delta x = q(x + 1, y) - q(x, y)$$

$$\Delta y = q(x, y + 1) - q(x, y)$$

De esta forma, los componentes del pseudogradiente pueden ser encontrados considerando las diferencias Δx y Δy por el número de píxeles entre dos puntos. De donde se puede encontrar la siguiente máscara de convolución:

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

$$|\nabla q| = |\nabla_x| + |\nabla_y| \approx |\Delta x| + |\Delta y| = |q(x, y) - q(x, y + 1)| + |q(x, y) - q(x + 1, y)|$$

Así, una forma simple, de aproximación de la ecuación 4.10, es obtener la diferencia $q_5 - q_8$ para distinguir los cambios bruscos en una imagen en la dirección x , $q_5 - q_6$ para ubicar los cambios en la dirección y y combinarlas en la ecuación:

$$|\nabla f| \approx \sqrt{(q_5 - q_8)^2 + (q_5 - q_6)^2}$$

$$\nabla f \approx |q_5 - q_8| + |q_5 - q_6|$$

Esta forma de operador es conocido como operador de Robert y fue uno de los primeros operadores usados para detectar bordes en imágenes. Este puede ser implementado mediante las siguientes máscaras de convolución:

1	0
-1	0

Vertical

1	-1
0	0

Horizontal

y empleando máscaras de 3x3:

0	0	0
0	1	0
0	-1	0

Vertical

0	0	0
0	1	0
0	-1	0

Horizontal

En lugar de encontrar las componentes aproximadas del gradiente a lo largo de las direcciones x y y , se pueden también aproximar los valores de los componentes del gradiente a lo largo de las direcciones a 45° y a -45° grados con respecto al eje x , resaltando los bordes en estas direcciones. En este caso se emplean las siguientes ecuaciones:

$$|\nabla f| \approx \sqrt{(q_5 - q_9)^2 + (q_6 - q_8)^2}$$

$$|\nabla f| \approx |q_5 - q_9| + |q_6 - q_8|$$

de donde se obtienen las siguientes máscaras:

1	0
0	-1

45°

0	1
-1	0

-45°

y empleando máscaras de 3×3 , puede ser modificado ligeramente para obtener:

0	0	0
0	1	0
0	0	-1

y

0	0	0
0	1	0
-1	0	0

Se puede observar que el operador de diferenciación se comporta como un filtro de alta frecuencia, por consiguiente, enfatiza los detalles de la imagen. Una desventaja inmediata, es en consecuencia su sensibilidad al ruido. Cuando se aplica a imágenes granuladas o texturizadas, la imagen resultante consiste de altas intensidades correspondientes tanto a los bordes verdaderos debidos a transiciones de intensidad, como a ruido (aspecto del grano, texturas, etc.). Esta sensibilidad al ruido puede ser reducida usando máscaras de mayor dimensión para aproximar el operador de diferenciación, tales como las máscaras que se presentan a continuación [1]. Es de anotar que las imágenes resultantes consisten de intensidades negativas y positivas, por lo tanto para su visualización se toma la magnitud del gradiente.

B. Máscara de Prewitt

Otra aproximación de la ecuación 4.10, se logra haciendo la diferencia en la siguiente forma:

$$|\nabla f| \approx |p_7 + p_8 + p_9 - p_1 + p_2 + p_3| + |p_3 + p_6 + p_9 - p_1 - p_4 - p_7|$$

de donde se obtienen las máscaras sugeridas por Prewitt:

-1	-1	-1
0	0	0
1	1	1

y

-1	0	1
-1	0	1
-1	0	1

Este operador realiza una detección horizontal y vertical e introduce igualmente un efecto de suavización para reducir los efectos de ruido o de texturas poco marcadas. De esta forma se puede realizar un efecto combinado en un solo operador.

El efecto de diferenciación viene de las partes positivas y negativas de la máscara que se obtienen de las anteriores ecuaciones aplicadas sobre los pixeles. El efecto de suavización se obtiene de la separación de un pixel existente en la máscara entre las partes positivas y negativas, lo cual genera un promedio entre estos valores.

En las imágenes de la figura 4.28 se aprecia el resultado de aplicar estas máscaras sobre la imagen de Lena.



Figura 4.28: Imágenes tratadas con las máscaras de Prewitt en x y en y .

Igualmente puede ser empleada una malla hexagonal [1]:

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

C. Otras máscaras lineales

Muchos detectores de borde se han diseñado usando las técnicas de máscara de convolución, usando frecuentemente tamaños de máscaras más grandes. Una ventaja de usar un tamaño más grande de máscara es que los errores debido al ruido son reducidos al realizar un promedio local entre los pixeles que se encuentran al interior de la máscara, obteniendo de esta manera un efecto de suavización (filtro pasa-bajo). Se usan también normalmente máscaras de tamaño impar, dado que los operadores se pueden centrar alrededor del pixel de interés y por lo tanto, puede proveer una mejor estimación del gradiente en el punto [33].

Se tienen diferentes combinaciones de valores que llevan a realizar en una imagen una detección de contornos horizontales, así:

0	0	0
-1	0	1
0	0	0

Del mismo modo que para la detección horizontal se tienen las siguientes combinaciones de valores dentro de la máscara para evaluar los cambios en la dirección vertical:

0	1	0
0	0	0
0	-1	0

D. Operador de Sobel

Este operador, no lineal, introducido por Sobel realiza una detección horizontal y vertical e introduce también un efecto de suavización. Las máscaras del operador se representan de la siguiente manera:

$$\nabla_x = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \quad \text{y} \quad \nabla_y = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

Como se observa en las máscaras, los coeficientes son escogidos de manera que crean combinaciones no lineales sobre los 9 pixeles sobre los cuales se aplica la máscara:

$$\nabla_x = (p_7 + 2p_8 + p_9) - (p_1 + 2p_2 + p_3)$$

$$\nabla_y = (p_3 + p_6 + p_9) - (p_1 + 2p_4 + p_7)$$

Un ejemplo de las imágenes obtenidas luego de la convolución con este operador puede ser visto en la figura 4.29.

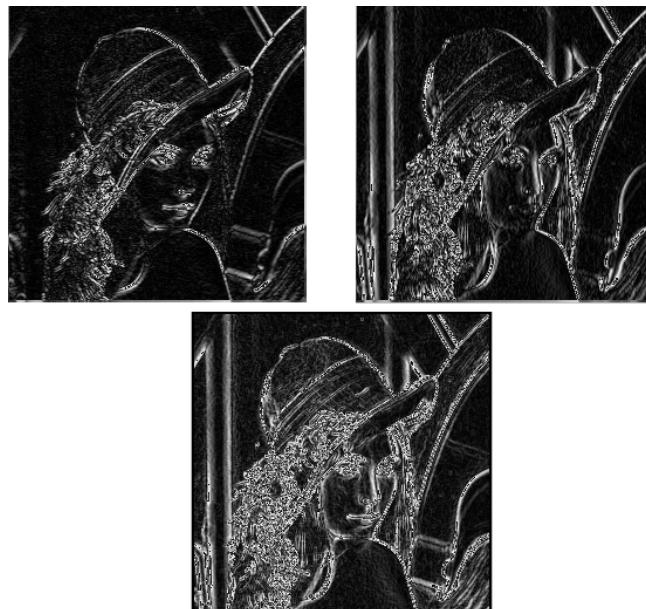


Figura 4.29: Imagen tratada con Sobel horizontal, vertical y total.

Una malla ortogonal de Sobel está dada por [1]:

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

E. Otras máscaras no lineales

Las siguientes máscaras intensifican, de acuerdo a los valores dados, los contornos que lleven la dirección específica que detecta cada máscara.

a. Detección horizontal Se tienen diferentes combinaciones de valores que llevan a realizar en una imagen una detección de contornos horizontales, así:

-1	2	-1
-1	2	-1
-1	2	-1

b. Detección vertical Del mismo modo que para la detección horizontal se tienen las siguientes combinaciones de valores dentro de la matriz para evaluar las imágenes:

-1	-1	-1
2	2	2
-1	-1	-1

c. Detección oblicua Los contornos de las imágenes que lleven esta dirección pueden ser detectados por alguna de estas máscaras:

$$\begin{array}{|c|c|c|} \hline -1 & -1 & 2 \\ \hline -1 & 2 & -1 \\ \hline 2 & -1 & -1 \\ \hline \end{array} \quad y \quad \begin{array}{|c|c|c|} \hline 2 & -1 & -1 \\ \hline -1 & 2 & -1 \\ \hline -1 & -1 & 2 \\ \hline \end{array}$$

d. Máscara no lineal de Prewitt Entre otras máscaras no lineales, encontramos una presentada en [1] a partir de las máscaras de Prewitt:

$$\begin{array}{|c|c|c|} \hline 1 & 1 & -1 \\ \hline 1 & -2 & -1 \\ \hline 1 & 1 & -1 \\ \hline \end{array} \quad y \quad \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & -2 & 1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Otra serie de máscaras han sido propuestas por diferentes compañías productoras de software, así se tienen por ejemplo las siguientes máscaras normalizadas:

$$h_{Pasaaaltos}(i, j) = \frac{1}{b+2} \begin{bmatrix} 1 & -b & 1 \\ -b & b^2 & -b \\ 1 & -b & 1 \end{bmatrix}$$

F. Máscaras circulares Otra forma de implementar máscaras derivativas, consiste en el empleo de las que pueden denominarse máscaras circulares. El proceso consiste en la utilización de una máscara, que se rota un número de veces igual a su tamaño menos uno para el caso de las máscaras cuadradas. Cada vez que se rota la máscara, se aplica a la imagen original como una máscara sencilla. Al término del proceso se toma como valor final de gradiente para cada pixel el mayor entre los resultados obtenidos al aplicar cada rotación de la máscara a la imagen como se vera más adelante.

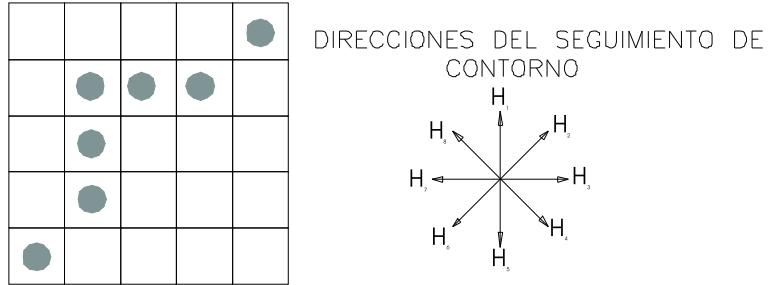
a. Máscaras de Freeman Freeman desarrolló un algoritmo para realizar la compresión de imágenes, de manera que de la imagen se guardan únicamente las direcciones que siguen los pixeles del contorno del objeto de interés. Por ejemplo, en la figura 4.30, partiendo del pixel del extremo inferior izquierdo, se puede obtener la secuencia mostrada, tomando en cuenta las 8 posibles direcciones a las que se puede ir a partir de un pixel dado.

El cálculo del gradiente es aproximado, mediante un operador consistente de ocho máscaras, cada una de las cuales corresponde a una dirección preferencial obtenida mediante una rotación de 45 grados entre máscara y máscara. Cada una de las 8 máscaras evalúan una dirección y permiten la codificación directa de los contornos siguiendo el método de Freeman:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}_{h_0} \quad \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}_{h_1} \quad \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}_{h_2} \quad \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{h_3}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{h_4} \quad \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}_{h_5} \quad \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}_{h_6} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}_{h_7}$$

El pseudo-gradiente $\nabla(x, y)$ de la imagen p está dado por:



De acuerdo a la anterior regla la imagen mostrada presentara un sistema de compresion asi:

2 1 1 3 3 2 etc...

Figura 4.30: Seguimiento de contorno por el método de Freeman.

$$\nabla(x, y) = \max_{i=0}^7 \{h_i * p\}$$

donde el símbolo $*$ representa el operador de convolución y la orientación se obtiene a partir de:

$$45 \times \arg \max_{i=0}^7 |h_i * p|$$

b. Máscaras de Kirsh En estas máscaras el pseudo-gradiante ∇ es calculado, escogiendo el valor máximo de acuerdo a la expresión:

$$\nabla = \max_{i=0}^7 [5S_i - 3T_i]$$

donde, basada en la siguiente nomenclatura para una serie de píxeles en la imagen:

$$\begin{bmatrix} p_0 & p_1 & p_2 \\ p_7 & & p_3 \\ p_6 & p_5 & p_4 \end{bmatrix}$$

se definen:

$$S_i = p_i + p_{i+1} + p_{i+2}$$

$$T_i = p_{i+3} + p_{i+4} + p_{i+5} + p_{i+6} + p_{i+7}$$

siendo los subíndices de p_i evaluados en modulo 8 [41].

Este operador permite obtener la amplitud máxima del gradiente, considerando el barrido de los píxeles siguiendo el sentido de las manecillas del reloj, alrededor del pixel en estudio. Las máscaras de Kirsh son entonces las siguientes:

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

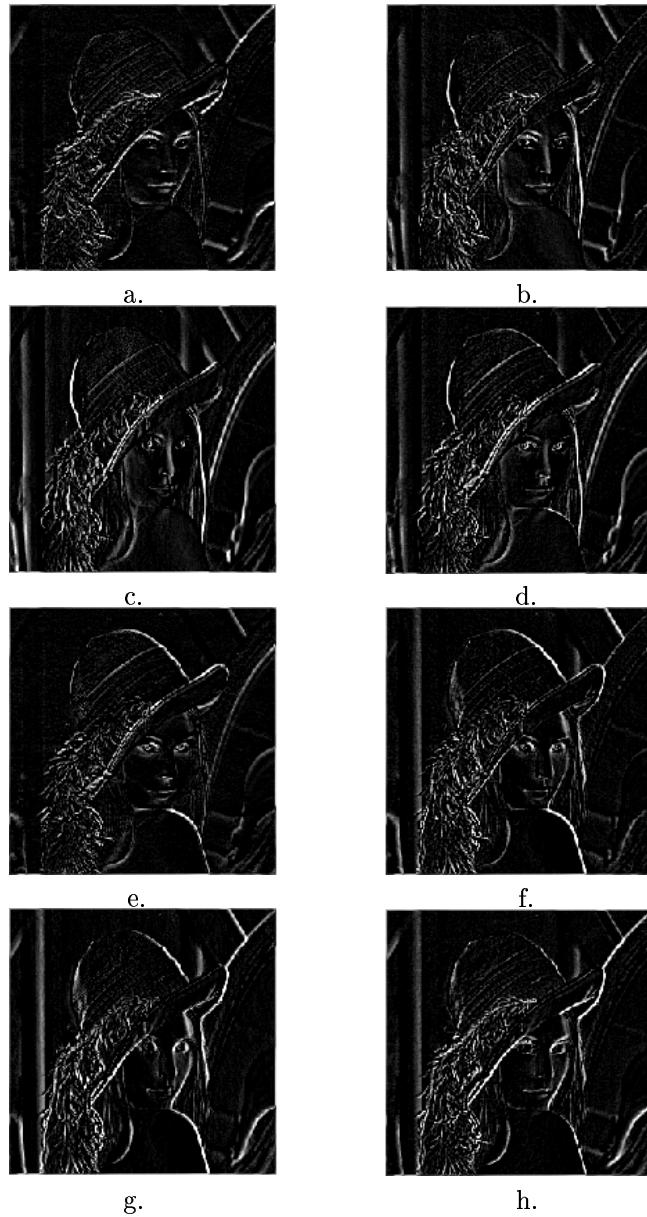


Figura 4.31: Imágenes tratadas con máscaras de Freeman: a. h_1 . b. h_2 . c. h_3 . d. h_4 . e. h_5 . f. h_6 . g. h_7 . h. h_8 .

c. Generalización El uso de máscaras circulares puede ser generalizado para las otras máscaras vistas. Dado que el gradiente es la derivada de la función del nivel de gris y corresponde a la pendiente de función en cada punto de la imagen, puede ser calculado en cada una de las principales direcciones de la máscara, lo que es realizado rotando el operador en todas las direcciones posibles. Por ejemplo, en caso de emplear las máscaras cuadradas de Prewitt, las siguientes máscaras son obtenidas:

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

 $\uparrow N$ $\nearrow NE$ $\rightarrow E$ $\searrow SE$

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

 $\downarrow S$ $\swarrow SO$ $\leftarrow O$ $\nwarrow NO$

y en el caso de máscaras hexagonales:

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

 $\rightarrow 0^\circ$ $\nearrow 60^\circ$ $\nwarrow 120^\circ$ $\leftarrow 180^\circ$ $\swarrow 240^\circ$ $\searrow 300^\circ$

La amplitud del gradiente de la imagen está dada por el máximo de los valores absolutos de los diferentes gradientes en todas las direcciones posibles.

La orientación del gradiente se obtiene de la dirección preponderante del gradiente en cada punto, el cual corresponde al valor del gradiente de la imagen. Los niveles de gris caen entre 1 y 8 para una máscara cuadrada de tamaño 3x3 (las 8 direcciones entre 0° y 315°) y 1 a 6 para una máscara hexagonal.

4.9.12 Umbralización del gradiente

Una normalización de la imagen resultante luego de aplicar el operador gradiente, puede ser realizada al dividir el valor del gradiente entre la suma de los valores absolutos de las máscaras, lo cual reduce la sensibilidad al ruido y evita el desbordamiento, pero disminuye el contraste. Solo los bordes principales aparecerán en la imagen resultante. Si no se realiza una normalización, se debe tener cuidado con un posible desbordamiento, pero los bordes serán mucho más evidentes.

La imagen resultante al obtener la magnitud de gradiente, luego de convolucionar la imagen original con un operador de gradiente, es una imagen en niveles de gris, cuyos bordes son enfatizados en dos direcciones ortogonales. El siguiente paso en la detección de los bordes es identificar los pixeles que realmente corresponden a puntos de los bordes. En la imagen resultante, un borde está formado por las líneas formadas por las crestas del gradiente. Por lo tanto, los bordes pueden ser definidos como aquellos puntos cuya magnitud de gradiente es máxima a lo largo de la dirección del gradiente, es decir la dirección perpendicular al borde. Este método, al conservar los máximos locales, provee líneas finas de los contornos. Una técnica bastante común consiste en identificar como pixeles del borde, sólo aquellos puntos cuya magnitud del gradiente sea suficientemente grande, mayor que un umbral, es decir aquellos que indican una transición brusca de la intensidad o nivel de gris.

4.9.13 Comparación entre las máscaras de gradiente

Los primeros detectores trataron de resolver básicamente dos problemas, primero calcular algunas derivadas de la intensidad de la imagen y, segundo, ser robustas al ruido. Estos dos requerimientos son contradictorios y varias propuestas, como se ve más adelante, han sido hechas para alcanzar un balance entre la detección precisa de los bordes y la robustez al ruido.

La idea de medir la calidad de un detector es relativamente reciente, en 1978, Abdou y Pratt definieron una figura de mérito para comparar los diferentes detectores de borde. Ella pretende analizar el número de píxeles ideales que debe llevar el contorno de la imagen y el número real de píxeles que lleva el contorno obtenido:

$$F = \frac{1}{\max(I_i, I_A)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d^2(i)}$$

donde:

- I_i = número de puntos del contorno ideal.
- I_A = número de puntos del contorno real.
- $d(i)$ = distancia más corta entre el i -ésimo punto real del borde y el punto del contorno ideal.
- α = constante positiva.

$F \leq 1$, siendo $F = 1$ cuando $I_i = I_A$ y $d(i) = 0 \forall i$ [17].

El uso de una medida de la calidad de un detector de borde se ha mostrado bastante útil. La figura 4.32 muestra el resultado al emplear la figura de Abdou y Pratt para comparar algunas máscaras.

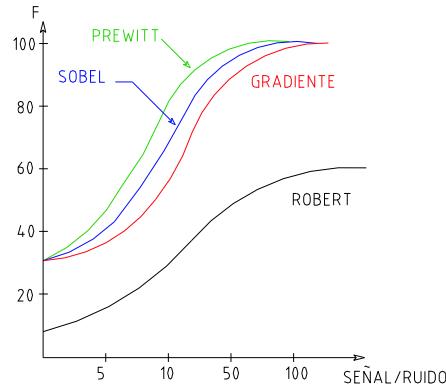


Figura 4.32: Comparación entre las distintas máscaras para el cálculo del gradiente.

4.9.14 Laplaciano

Todos los detectores de borde vistos hasta ahora aproximan las derivadas de primer orden de los valores de los píxeles de una imagen. Es posible también usar las derivadas de segundo orden para la detección de bordes. Un operador muy popular de segundo orden es el Laplaciano. Si se desarrollan las derivadas parciales $\frac{\partial f}{\partial x}$ y $\frac{\partial f}{\partial y}$ de segundo orden de una función continua $f(x, y)$, se puede encontrar la expresión del operador laplaciano denotado por $\nabla^2 f(x, y)$ ó $\Delta f(x, y)$ como:

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

Nuevamente se puede usar una aproximación de diferencia discreta para estimar las derivadas y representar el operador laplaciano por medio de máscaras de convolución de dimensión 3x3.

$$\frac{\partial^2 f(x, y)}{\partial x^2} \approx [f(x, y) - f(x-1, y)] - [f(x+1, y) - f(x, y)] = -f(x+1, y) + 2f(x, y) - f(x-1, y)$$

$$\frac{\partial^2 f(x, y)}{\partial y^2} \approx [f(x, y) - f(x, y - 1)] - [f(x, y + 1) - f(x, y)] = -f(x, y + 1) + 2f(x, y) - f(x, y - 1)$$

de donde se obtiene:

$$\nabla^2 f(x, y) = 4f(x, y) - f(x + 1, y) - f(x - 1, y) - f(x, y + 1) - f(x, y - 1)$$

Esta ecuación aplicada sobre el pixel resulta en:

$$\nabla^2 f \approx 4p_5 - (p_2 + p_4 + p_6 + p_8)$$

de donde se obtiene la siguiente máscara de convolución del operador laplaciano (ver ejemplo en la figura 4.33):

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Como este operador discreto no es realmente isotrópico, una máscara hexagonal puede ser implementada:

$$\begin{bmatrix} -1 & -1 \\ -1 & 6 & -1 \\ -1 & -1 \end{bmatrix}$$

La máscara resultante, como otras presentadas, resalta el contorno de una imagen dada. Este operador es no direccional y en consecuencia refuerza los bordes independientemente de su dirección y su pendiente. Es conveniente resaltar que la imagen obtenida es muy oscura y por lo tanto debe aumentarse su brillo y contraste a fin obtener un resultado de acuerdo al tratamiento de la imagen que se requiere.



Figura 4.33: Imagen resultante luego de ser tratada con Laplaciano.

Sin embargo este operador presenta algunas desventajas:

El operador gradiente aumenta los efectos de ruido y el operador laplaciano los aumenta aún más, debido a que la derivada de segundo orden es más sensible al ruido que la derivada de primer orden. Esto hace que la distinción entre los bordes y el ruido sea más difícil en el laplaciano que en el gradiente.

Además, como el laplaciano no es un vector, no se obtiene información acerca de la dirección en la cual se encuentra el borde. Los problemas que la presencia de ruido ocasiona, se pueden atenuar aplicando antes o simultáneamente con el proceso de detección de borde alguna técnica de reducción de ruido como las expuestas anteriormente.

Una normalización se puede hacer al dividir el valor del laplaciano por la suma de los valores absolutos de las máscaras. Esto reduce la sensibilidad al ruido, evita el rebosamiento, pero disminuye el contraste. Sólo los bordes principales aparecerán sobre el laplaciano de la imagen. Cuando no se realiza ninguna normalización puede haber rebosamiento, pero los bordes serán mucho más evidentes.

Para remover el ruido, se suaviza la imagen primero o se puede emplear una técnica de detección de cruces por cero del laplaciano. Los puntos de cruces por cero son aquellos sitios en los cuales el laplaciano cambia de negativo a positivo y viceversa.

El laplaciano construido con los 8 vecinos (ver filtro pasa-altos) es más preciso, los bordes son mejor detectados, pero los puntos aislados son más numerosos. El laplaciano construido con 4 vecinos es menos preciso pero también menos sensible al ruido. Además también favorece las direcciones horizontales y verticales [1].

Comparada con la técnica de umbralización del gradiente, la detección de cruces por cero del laplaciano entrega curvas cerradas de un pixel de ancho; lo cual es importante cuando se desea que la localización del borde sea precisa.

Capítulo 5

Segmentación por bordes

La segmentación de imágenes es un proceso por el cual una imagen se subdivide o partitiona en zonas o clases disyuntas con el propósito de identificar estructuras u objetos presentes en la imagen. En términos formales, la segmentación s puede definirse como una operación de mapeo desde una imagen o conjunto de $M \times N$ pixeles hacia un conjunto finito de clases C , es decir:

$$s : M \times N \rightarrow C$$

donde:

$$C = \{1, 2, \dots, c | 2 \leq c \leq k\}$$

El número de clases k está determinado por el dominio del problema y la meta de operación de la segmentación.

Haralick y Shapiro [21] proponen las siguientes características para una buena segmentación:

- Las regiones resultantes del proceso de segmentación deben ser uniformes y homogéneas con respecto a alguna característica de la imagen, tal como el nivel de gris, la textura, etc.
- Las regiones internas deben ser simples y no incluir huecos o estructuras ruidosas.
- Las regiones adyacentes deben tener valores diferentes con respecto a la característica según la cual son uniformes.
- Las fronteras o límites de cada segmento deben ser lo más simples posible.

5.1 Segmentación basada en la obtención de bordes

Dado que el objetivo de la segmentación es clasificar y separar los objetos presentes en la imagen, utilizando para ellos atributos o características que permitan discriminálos, resulta lógico delimitar la frontera o borde de cada uno de ellos como un paso para el cálculo de sus atributos discriminantes.

El cálculo de extracción de bordes de los objetos puede plantearse como una operación de filtrado de la imagen original. Con este fin se emplea un filtro pasaaltos o de gradiente. Dado que las técnicas de realce de bordes favorecen las frecuencias espaciales elevadas, debe tomarse en cuenta que además de reforzar los bordes, también se refuerzan bordes inexistentes debido a la presencia de ruido. Por lo tanto, existe un compromiso entre la magnitud del realce del borde deseado y el grado de amplificación del ruido que puede ser tolerado.

Las técnicas de segmentación por detección de bordes se realizan básicamente en dos etapas diferentes:

- La primera corresponde a la determinación en la imagen de los pixeles que probablemente pertenecen a los bordes del objeto de interés por medio de la observación de las discontinuidades en el gradiente o en los pasajes por cero del laplaciano de la imagen.
- La segunda etapa busca unir los bordes con el objetivo de obtener un contorno continuo.

5.2 Métodos avanzados de segmentación de imágenes por bordes

Además de los métodos de obtención de bordes de los objetos mediante el empleo de máscaras de convolución que permiten calcular un pseudo-gradiente y el laplaciano, existen otra clase de métodos introducidos recientemente.

Los contornos de los objetos en las imágenes corresponden en general a los máximos locales del gradiente o a los ceros del laplaciano de la función de nivel de gris. Las dificultades en la detección del contorno provienen del ruido presente en las imágenes. Se tiene en consecuencia, el problema de diferenciar el contorno del objeto dentro de una señal ruidosa. Para resolverlo se siguen los siguientes pasos:

1. Se definen unos criterios de ejecución de un operador incluyendo la modelización de los contornos buscados y la noción de ruido.
2. A partir de los criterios se deduce una familia de filtros optimales.
3. Se resuelve el problema dado por la implantación de estos filtros.

A continuación se estudia la utilización de filtros lineales. Estos tienen la ventaja, cuando no se tiene información apriori de la imagen, de proveer algoritmos de baja complejidad y resultados satisfactorios en la mayoría de los casos. Se estudian particularmente los filtros separables recursivos.

Los filtros separables tienen varias ventajas [34]:

- Pueden generalizarse directamente a cualquier dimensión.
- Reducción del tiempo de cálculo. Así, para el caso de una imagen bidimensional de tamaño $p \times p$, una operación de convolución pasa de p^2 a una $2p$.
- Posibilidad de tomar en cuenta que el ruido en las imágenes puede tener características diferentes según su dirección.
- Es posible la elaboración de algoritmos recursivos, los cuales permiten una implantación de bajo costo algorítmico para operadores de respuesta impulsional infinita.

La mayor desventaja es que puede llegar a obtener filtros anisotrópicos, los cuales varían según la dirección tomada siguiendo una dirección diferente de los ejes x y y , donde la anisotropía es definida con relación a la distancia euclíadiana. Esto provoca una ligera delocalización de los contornos obtenidos. Varios ejemplos de filtros anisotrópicos [19] han aparecido en la literatura que pueden resultar de interés, pero no son el objetivo de este documento. Se pueden mencionar los filtros gaussianos entre los raros filtros separables e isotrópicos.

Para la detección de contornos se determinan pues, filtros lineales que permiten aproximar el gradiente o el laplaciano de la imagen.

Sea un filtro lineal de respuesta impulsional h y f una señal, la derivada de la señal convolucionada con f es igual a la imagen convolucionada con la derivada de f [34]:

$$(f * h)' = f * h'$$

$$(f * h)'' = f * h''$$

En el caso donde h es un filtro separable:

$$f * h(x, y) = f * [h_1(x)h_2(y)]$$

Es decir, que la convolución de una señal por el producto de dos respuestas impulsionales separables, puede escribirse como la convolución por la segunda respuesta de la señal convolucionada con la primera respuesta:

$$f * [h_1(x)h_2(y)] = [f * h_1(x)] * h_2(y)$$

Suponiendo un ruido homogéneo en todas las direcciones se puede afirmar que:

$$h_1 = h_2 = h$$

Por facilidad, se supondrá a partir de ahora que el ruido es isotrópico. Sin embargo, el acercamiento siguiente se adapta directamente al caso de un ruido anisotrópico pero homogéneo según las direcciones x y y .

Normalmente en los filtros a estudiar se ha tomado como entrada una función paso, para resolver ecuaciones de convolución:

$$q(x) = u * f = \int u(x-y)f(y)dy$$

donde $u(x-y)$ es la función paso.

La regularización es obtenida filtrando $q(x)$ con un filtro de suavización pasabajo y la diferenciación es obtenida sobre la versión filtrada de $q(x)$.

Criterios de rendimiento

En general, el resultado de la detección de un borde en una imagen es difícil de juzgar debido a varias causas:

- El número y la clase de imágenes que pueden ser tratadas son, a menudo, bastante limitadas.
- Frecuentemente los resultados son evaluados subjetivamente, según la apreciación de las personas, lo cual no permite obtener un juzgamiento objetivo.

Por esta razón, es necesario definir algunos criterios de rendimiento de un detector de borde según un modelo de contorno y de ruido escogidos.

5.2.1 Operador de Marr y Hildreth

Marr y Hildreth se basaron en la teoría conocida de la visión en los seres humanos y mamíferos para construir un modelo que pudiera ser empleado en visión artificial [38]. Ellos encuentran que el propósito del procesamiento visual inicial consiste en construir una descripción primitiva pero rica de la imagen. La descripción más simple es llamada un boceto primario (en inglés: primal sketch), compuesto en su mayoría por bordes.

Este estudio puede ser sintetizado en los siguientes puntos [38]:

- Las características de interés en las imágenes naturales ocurren en diferentes escalas o niveles de definición. Ningún operador sencillo puede detectar simultáneamente todas las distintas escalas de definición de los bordes. Por lo tanto, varios operadores, cada uno trabajando a una escala de definición diferente, deben ser aplicados sobre la imagen original y los obtenidos deben ser combinados.
- Aparentemente una imagen natural no consiste de patrones de difracción ni otros efectos ondulatorios, por lo que algún tipo de promedio local o suavización debe llevarse a cabo para permitir el proceso de la visión.

Por lo tanto, un prefiltrado debe ser aplicado antes de la diferenciación. Este debería estar localizado tanto en el dominio espacial (suavización) como frecuencial (suavización y de banda limitada), algo que no es posible, sobre la base de que los bordes son características de las imágenes que están localizadas tanto espacial (características locales) como frecuencialmente, dado que ellas ocurren a cierta escala. Sin embargo la función Gaussiana, ya vista antes, es la que más se acerca a este criterio [17][38]:

$$G_\sigma(r) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}$$

donde $r^2 = x^2 + y^2$.

Este es un buen ejemplo de una respuesta impulsiva a la vez isotrópica (función de r solamente) y separable (el producto de una función de x y una función de y).

Para detectar los cambios de intensidad en una imagen $q(x, y)$, ésta es pretratada efectuando una suavización con un filtro gaussiano $G_\sigma(x, y)$. Enseguida, se calcula la derivada segunda en la dirección del gradiente θ y se buscan los pasajes por cero en la imagen resultante q . Es decir, se buscan los cruces por cero en:

$$q(x, y) = \frac{\partial^2[G_\sigma(r) * p(x, y)]}{\partial\theta^2}$$

En una dimensión la derivada segunda de la función de Gauss, mostrada en la figura 5.1.a, está dada por:

$$\frac{\partial^2 G_\sigma(x)}{\partial x^2} = \frac{1}{\sqrt{2\pi}\sigma^3} \left(\frac{x^2}{\sigma^2} - 1 \right) e^{-\frac{x^2}{2\sigma^2}}$$

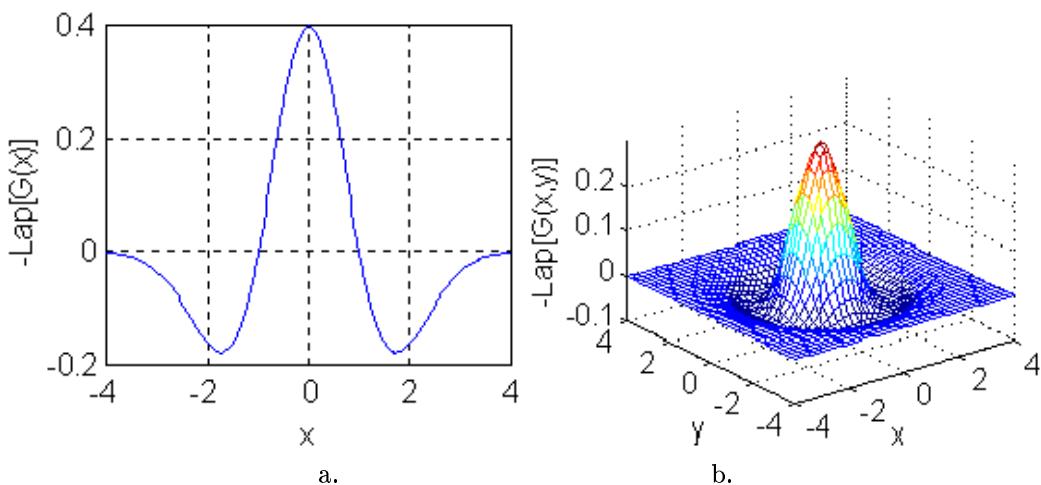


Figura 5.1: Derivada segunda de la función de Gauss: a. En una dimensión. b. En dos dimensiones.

Como se observa en la figura 5.1.a, la función $-\frac{\partial^2 G_\sigma(x)}{\partial x^2}$, presenta la forma característica de una función pasaaltos (ver figura 4.11).

La figura 5.1.b muestra la derivada segunda de la función de Gauss en dos dimensiones. Puede verificarse a partir del teorema de derivación de una convolución que:

$$\frac{\partial^2[G_\sigma(r) * p(x, y)]}{\partial\theta^2} = \frac{\partial^2G_\sigma(r)}{\partial\theta^2} * p(x, y) \quad (5.1)$$

Se puede suponer que los pasajes por cero de la derivada segunda direccional coinciden con aquellos del laplaciano, así esta operación puede ser simplificada convolucionando la imagen original $f(x, y)$ con un operador Laplaciano de un Gaussiano (*LOG*), el cual es mostrado en la figura XXX y la expresión 5.1 llega a ser:

$$q(x, y) = \nabla^2(G_\sigma * p(x, y)) = \Delta(G_\sigma * p(x, y))$$

Un ejemplo se presenta en la figura 5.2. El prefiltrado de la imagen con un filtro pasabajos es importante ya que el laplaciano es infinito para una función paso no suavizada y la posición del borde se preserva luego del filtrado.

Debe tenerse en cuenta que el operador *LOG* es aún susceptible al ruido, pero sus efectos pueden ser reducidos ignorando los cruces por cero producidos por pequeños cambios en la intensidad de imagen. Además



Figura 5.2: Imagen filtrada con el operador LOG con $\sigma = 1$.

de la posición del borde, este operador permite conocer su dirección determinada a partir del cruce por cero [33].

Otro método relacionado emplea el operador llamado Diferencia del Gaussiano *DOG*. En él se aplican dos operadores gaussianos con diferente valor de σ a una sola imagen y luego se determina la diferencia entre las dos imágenes suavizadas [9]. El *DOG* no es un filtro separable. Sin embargo, es posible hacer una implementación eficiente, simplemente haciendo la substracción entre las imágenes resultantes de convolucionar la imagen original con dos máscaras gaussianas con dos valores de σ diferentes. Existen evidencias de que el sistema visual humano emplea un método similar [49].

Es decir, se obtiene el laplaciano de una gaussiana $\nabla^2 G_\sigma(x, y)$ que puede ser aproximado por la diferencia de 2 gaussianas (operador *DOG*). A continuación se demuestra que esta operación de diferencia se aproxima al operador *LOG*.

En el caso monodimensional:

$$DOG(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} - \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_1^2}}$$

Tomando $\sigma_1 = \sigma + \delta\sigma$, siendo δ un valor escogido por el usuario, entonces

$$\sqrt{2\pi}DOG(x) = \frac{1}{\sigma} e^{-\frac{x^2}{2\sigma^2}} - \frac{1}{\sigma + \delta\sigma} e^{-\frac{x^2}{2(\sigma + \delta\sigma)^2}}$$

$$\sqrt{2\pi}DOG(x) \approx \delta\sigma \frac{\partial}{\partial\sigma} \left[\frac{1}{\sigma} e^{-\frac{x^2}{2\sigma^2}} \right] = - \left[\frac{1}{\sigma^2} - \frac{x^2}{\sigma^4} \right] e^{-\frac{x^2}{2\sigma^2}}$$

Por lo tanto:

$$\left[\frac{1}{\sigma^2} - \frac{x^2}{\sigma^4} \right] e^{-\frac{x^2}{2\sigma^2}} = \frac{\partial^2 G(x)}{\partial x^2}$$

$DOG(x)$ aproxima la derivada segunda de $g(x)$ si σ y σ_1 son vecinos.

El objetivo es realizar un operador diferencial de banda estrecha. Estudiando el filtro *DOG*, se observa que la banda pasante es prácticamente constante para una relación $\frac{\sigma}{\sigma_1} < 1.6$. Sabiendo que la ganancia aumenta a medida que $\frac{\sigma}{\sigma_1}$ crece, se escoge $\frac{\sigma}{\sigma_1}$ vecino de 1.6 [9]. (Ver ejemplo en la figura 5.3).

La salida obtenida luego de aplicar un operador *DOG* no aporta una simplificación obvia de la imagen. Sin embargo, la información acerca de los bordes se encuentra en los sitios en donde los valores positivos y negativos son adyacentes, es decir, en los cruces por cero. La manera más efectiva para desplegar los cruces por cero, luego de una operación de convolución, es umbralizar la salida obtenida. El proceso de umbralización consiste en producir una imagen binaria, asignando un único valor a todos los pixeles que excedan un valor de umbral dado y otro valor a los demás. En general, la imagen binaria está constituida por pixeles negros y blancos. Para detectar los cruces por cero el valor de umbral se hace igual a cero. Sin embargo, ésta no es una muy buena técnica de segmentación de imágenes, excepto en situaciones controladas, donde se maneja la iluminación y el



Figura 5.3: Imagen filtrada con el operador de Marr-Hildreth con $\sigma = 0.5$ y $\sigma_1 = 1.5$.

contraste con el fondo y donde el ruido no es alto comparado con la señal. Si se asigna el color negro a los pixeles cuyo laplaciano es menor o igual a cero y el blanco cuando no lo es, en la imagen resultante los cruces por cero serán obviamente los bordes que separan las regiones negras de las blancas (ver figura 5.4.a). Filtrando la imagen original con la función LOG para diferentes valores de σ y luego realizando la detección de cruces por cero para cada una de las imágenes resultantes, puede observarse que para valores de σ más pequeños, la posición de los bordes es obtenida con mayor exactitud y se observan más detalles de la textura (ver figura 5.4.a). Para valores de σ mayores (ver figuras 5.4.b y 5.4.c) se conservan las características principales de la imagen, pero sólo se obtiene una aproximación de la posición de los bordes.

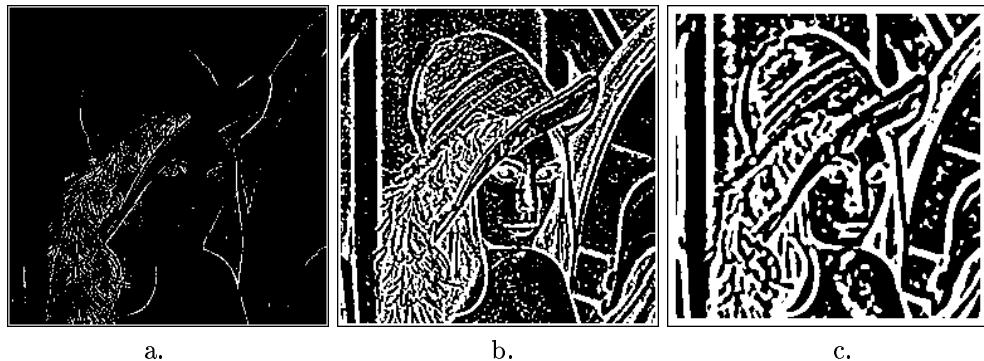


Figura 5.4: Imagen resultante luego de aplicar un filtro *LOG* y binarizar con: a. $\sigma = 0.5$. b. $\sigma = 1$. c. $\sigma = 2$.

Marr y Hildreth propusieron que los sistemas visuales biológicos combinan los valores obtenidos de cruces por cero de las imágenes filtradas con diferentes valores de σ para obtener evidencia de los bordes más significativos. Aunque aún esté sujeto a duda, es claro que un proceso análogo al realizado con el filtro *DOG*, se sucede en las etapas tempranas de la visión biológica. Los resultados de estas operaciones deben ser, por consiguiente, una forma eficiente para codificar la estructura de una imagen, dado que los mecanismos de la retina han evolucionado para transmitir esta clase particular de información a lo largo del nervio óptico. Esto es evidente al analizar, por ejemplo, una imagen en niveles de gris donde existe una gran cantidad de información redundante acerca del brillo de los objetos; ésta es sacrificada luego de un filtraje con el operador *DOG*, pero se conoce más acerca de la textura y los bordes [49].

Aunque el operador *DOG* es poco usado en detección de bordes, Young comenta que este se ha mostrado más valioso en algoritmos rápidos de encaje estereó (en inglés matching), donde se emplean imágenes umbralizadas luego de aplicar el *DOG* con diferentes valores de σ , para generar las regiones a encajar. La idea de usar los cruces por cero para distintos valores de σ se ha generalizado en la idea del análisis del estado del espacio [49].

5.2.2 Aproximación de Canny

El primero en proponer expresiones analíticas para definir algunos criterios para la detección de contornos fue John Canny en 1986 [5]. La aproximación de Canny fue desarrollada en el caso monodimensional [7]. Canny parte de un modelo en el espacio continuo, asumiendo como entrada del filtro un borde de entrada escalón sujeto a un ruido blanco gausiano, es decir supone un borde en $x = 0$. Así, sea $p(x)$ una señal monodimensional de entrada, representando un salto de amplitud U_0 , con ruido blanco estacionario $n(x)$ aditivo de media nula, desviación estandar n_0 :

$$p(x) = U_0 u(t) + n(x)$$

donde

$$U_0(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & x < 0 \end{cases}$$

con dsp (densidad espectral de potencia) $E[n^2(x)] = n_0 \forall x$.

Canny asume que la detección de bordes puede ser realizada convolucionando en el dominio continuo, la señal $p(x)$ con una función espacial $h(x)$, cuya amplitud es cero fuera del rango $[-W, W]$. Maximizando los bordes en la salida $q(x_0)$ de la convolución:

$$q(x_0) = p * h(x) = \int_{-w}^w p(x)h(x_0 - x)dx$$

Luego, el problema consiste en encontrar analíticamente el filtro óptimo de respuesta impulsional $h(x)$, tal que $q(x)$ sea máximo en el borde dado en el punto $x = 0$, que satisfaga los tres criterios de detección siguientes para la señal de entrada en escalón (ver figura 5.5).

Buena detección: El detector de borde debe responder sólo a estos y debe detectarlos a todos. Es importante que los bordes en la imagen sean detectados y no obtener bordes falsos.

Buena localización: La distancia entre los pixeles del borde hallados por el detector y el borde real debe ser tan pequeña como sea posible.

Respuesta única: El detector no debe identificar múltiples pixeles como pertenecientes a un borde, cuando sólo existe un único borde.

La respuesta del filtro $h(x)$ a la señal p en $x = 0$ está dada entonces, por la integral de convolución:

$$q(0) = \int_{-w}^w p(x)h(-x)dx$$

La función de transferencia del detector de borde $h(x)$ debe atenuar el ruido y localizar el borde. El borde estará definido por los máximos locales de la operación de gradiente $p(x) * h(x)$.

A. Criterio de buena detección

Se debe disminuir la probabilidad de no detectar bordes reales en la imagen y disminuir la probabilidad de detectar como bordes, puntos donde no existan contornos. Este criterio puede ser expresado por medio de la relación señal a ruido $\frac{S}{R}$, definida como el cociente entre la respuesta máxima debida a la señal de entrada solamente y la raíz cuadrada de la potencia de ruido en la salida [9]:

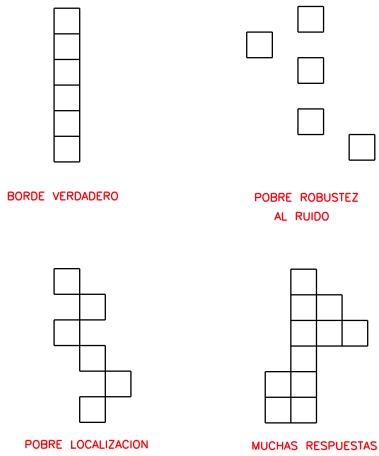


Figura 5.5: Criterios de detección del filtro de Canny (Adaptado de [17]).

$$\frac{S}{R} = \frac{U_0 \int_0^w h(x_0 - x) dx}{n_0 \sqrt{\int_{-w}^w \int_{-w}^w h^2(x) dx}}$$

Así, se busca maximizar la relación señal a ruido del gradiente, es decir aumentar la señal y disminuir el ruido. El filtro buscado es un derivador con el fin de obtener una salida nula cuando la señal de entrada sea constante. Canny escoge la función $h(x)$ antisimétrica, definiéndola impar. Para el análisis en $x = 0$, se obtiene:

$$\frac{S}{R} = \frac{U_0 \int_{-w}^0 h(x) dx}{n_0 \sqrt{\int_{-w}^w h^2(x) dx}} = \frac{U_0}{n_0} \Sigma$$

Luego encontrar la respuesta impulsiva $h(x)$ que maximice Σ equivale a hallar el mejor operador de solo detección.

B. Criterio de buena localización

Los puntos marcados como bordes por el operador deben estar tan cerca como sea posible del centro del verdadero borde. El valor de localización L corresponde entonces al reciproco de la distancia entre el borde localizado por el operador y el borde real:

$$L = \frac{U_0 |h'(0)|}{n_0 \sqrt{\int_{-w}^w h'^2(x) dx}} = \frac{U_0}{n_0} \Lambda(h')$$

de manera, que al hallar la respuesta impulsiva $h(x)$ que hace Λ tan grande como sea posible, corresponde a encontrar el operador óptimo de solo localización.

C. Criterio de unicidad de respuestas

El filtro debe permitir solo una única respuesta por cada borde verdadero en la imagen. Por lo tanto, se debe limitar el número de picos producidos en la salida como respuesta a un único borde ruidoso de manera que disminuya la probabilidad de definir más de un borde. Esta restricción se define mediante la distancia x_{\max} ; ésta representa la distancia entre dos máximos consecutivos de la respuesta del filtro h cuando solo hay ruido presente y significa esencialmente que h no tendrá muchas respuestas sobre un mismo borde en una sola región.

$$x_{\max} = 2\pi \left(\frac{\int_{-\infty}^{\infty} h'^2(x) dx}{\int_{-\infty}^{\infty} h''^2(x) dx} \right)^{\frac{1}{2}} = kW$$

Por lo tanto, con el fin de minimizar el número de respuestas en la salida, la distancia x_{\max} es fijada a una fracción k del factor del ancho del operador W . Este criterio equivale a limitar el número de máximos locales detectados como respuesta a un solo borde.

Con base en la definición de estos criterios, Canny intenta hallar un filtro $h(x)$ de respuesta impulsional finita de ancho W , que maximice el producto $\frac{S}{R}L$, sujeto a la restricción de no tener múltiples respuestas. El problema de maximizar $\frac{S}{R}L$, corresponde a hallar el máximo de $\Sigma\Lambda$, bajo el tercer criterio. El producto $\Sigma\Lambda$ es invariante a los cambios de amplitud U_0 del escalón, y a los cambios de escala. Esto último puede demostrarse haciendo

$$h_e(x) = h\left(\frac{x}{e}\right)$$

de donde se obtiene:

$$\Sigma_e = \sqrt{e}\Sigma$$

$$\Lambda_e = \frac{1}{\sqrt{e}}\Lambda$$

luego:

$$\Sigma_e \Lambda_e = \sqrt{e}\Sigma \frac{1}{\sqrt{e}}\Lambda = \Sigma\Lambda$$

La función $h(x)$ resulta demasiado compleja para ser resuelta analíticamente. Sin embargo, Canny desarrolla un acercamiento variacional para obtener una solución general en la forma de un filtro de respuesta impulsional finita $h(x)$ definido en el rango $[0, W]$ y presentando una pendiente s en el origen. Así, $h(x)$ se obtiene como solución de:

$$2h(x) - 2\lambda_1 h''(x) + 2\lambda_2 h'''(x) + \lambda_3 = 0 \quad (5.2)$$

donde

$$\lambda_2 - \frac{\lambda_1^2}{4} > 0$$

$$\alpha^2 - \omega\varpi^2 = \frac{\lambda_1}{2\lambda_2}$$

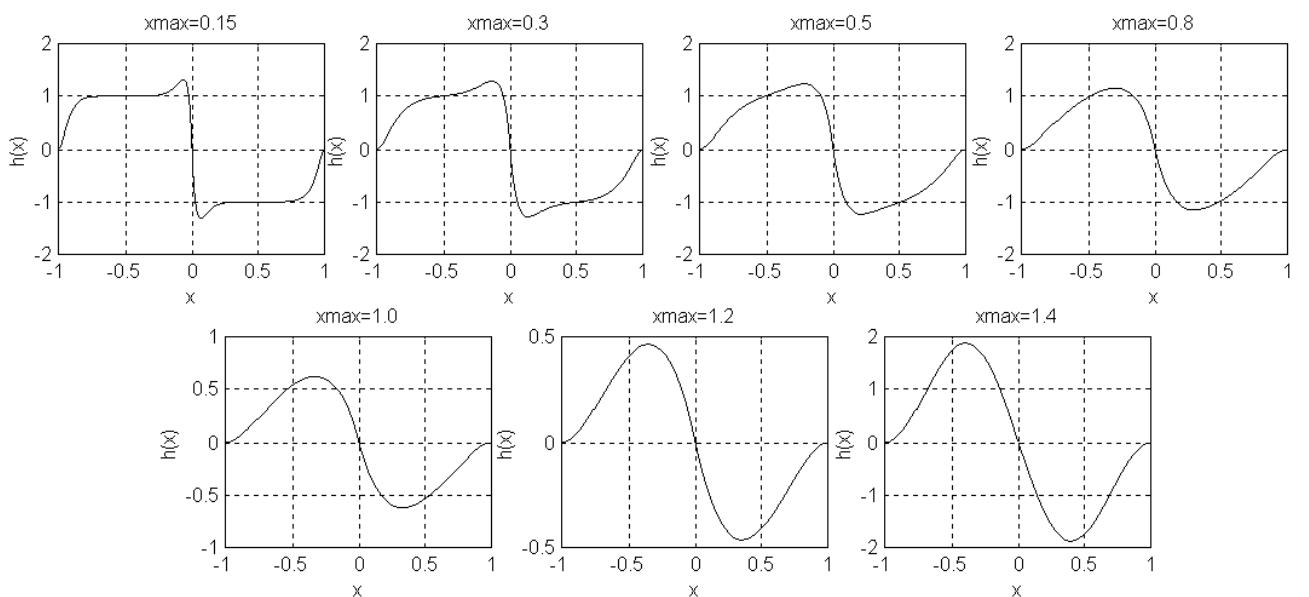
$$4\alpha^2\omega\varpi^2 = \frac{4\lambda_2 - \lambda_1^2}{4\lambda_2^2}$$

siendo los λ son los parámetros de ajuste.

La solución general puede ser entonces escrita como

Parámetros del filtro					
x_{\max}	$\Sigma \Lambda$	r	α	ω	β
0.15	4.21	0.215	24.59550	0.12250	63.97566
0.3	2.87	0.313	12.47120	0.38284	31.26860
0.5	2.13	0.417	7.85869	2.62856	18.28800
0.8	1.57	0.515	5.06500	2.56770	11.06100
1.0	1.33	0.561	3.45580	0.07161	4.80684
1.2	1.12	0.576	2.05220	1.56939	2.91540
1.4	0.75	0.484	0.00297	3.50350	7.47700

Tabla 5.1: Respuesta del operador de Canny para diferentes parámetros del filtro.

Figura 5.6: Operadores óptimos de Canny para $x_{\max} = 0.15, 0.3, 0.5, 0.8, 1.0, 1.2$ y 1.4 .

$$h(x) = e^{\alpha x} (a_1 \sin \omega x + a_2 \cos \omega x) + e^{-\alpha x} (a_3 \sin \omega x + a_4 \cos \omega x) + c$$

sujeto a las condiciones de límite:

$$h(0) = 0 ; h(W) = 0 ; h'(0) = s ; h'(W) = 0$$

Dado que la función $h(x)$ es impar, Canny extiende esta solución al rango $[-W, W]$ haciendo $h(x) = -h(-x)$. Las condiciones de límite establecidas se emplean para hallar los valores de a_1, a_2, a_3 y a_4 en términos de las constantes desconocidas α, ω, c y s .

Observando numericamente la respuesta del filtro para varios valores de x_{\max} (Tabla 5.1), Canny encuentra que el mejor filtro corresponde a los valores de $x_{\max} = 0.58\omega$, un valor de $\Sigma\Lambda = 1.12$. Este filtro es difícil de desarrollar. Sin embargo, Canny nota que la curva de respuesta del filtro (ver figura 5.6), en este caso, se aproxima mucho a la primera derivada de una función Gaussiana:

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

donde la derivada con respecto a x , mostrada en la figura 5.7, está dada por:

$$\frac{\partial G(x)}{\partial x} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

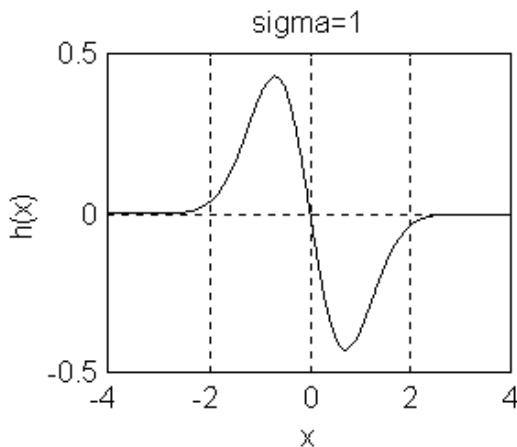


Figura 5.7: Función derivada de un gaussiano $\frac{\partial G(x)}{\partial x}$.

Esta función es mucho más fácil de calcular y puede ser extendida más eficientemente a dos dimensiones [Deriche]. El rendimiento de la primera derivada de una función Gaussiana es $\Sigma\Lambda \approx 0.92$. Así, esta solución presenta una desmejora con respecto al filtro óptimo en un valor cercano al 20%. Es de aclarar, como lo menciona Deriche [Deriche] que para la obtención de estos valores, Canny se basó únicamente en la observación comparativa de las curvas y ningún análisis comparativo de las aproximaciones ha sido hecho.

Canny estudia también la probabilidad de aparición de un máximo espurio en la vecindad de la posición exacta de la transición. Considerando que la respuesta a un escalón es prácticamente lineal en la vecindad de $x = 0$, se puede concluir que sólo habrá un máximo local cerca del centro del borde si la pendiente de la señal de salida es superior a la pendiente en la salida debida solamente al ruido. Sea P_m la probabilidad de este evento y P_f la probabilidad de marcar falsamente el borde, Canny establece la relación siguiente:

$$\frac{|h'(0)|}{\sqrt{\int_{-w}^w h''^2(x) dx}} = r \frac{\int_{-w}^0 h(x) dx}{\sqrt{\int_{-w}^w h^2(x) dx}}$$

donde r es una constante determinada por los valores de las probabilidades de P_s y P_f . r debe ser tan cercana como sea posible a 1 donde las probabilidades son identicas. Sin embargo, el filtro óptimo de Canny

debido a las restricciones tiene un r cercano a 0.58 y su aproximación, la primera derivada de una función gaussiana tiene un $r = 0.51$; lo cual corresponde a un deterioro de alrededor del 10%.

En el caso bidimensional, la función Gaussiana está dada por:

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

y la función se puede derivar, tanto con respecto a la dirección x , como con respecto a la dirección y . Así, convolucionando la imagen original con la función $\frac{\partial G(x, y)}{\partial x}$ y con $\frac{\partial G(x, y)}{\partial y}$, se obtiene una imagen resultante cuyos bordes son resaltados aún en presencia de ruido, el cual ha sido incorporado en el modelo de la imagen. El valor de σ se puede escoger siguiendo el análisis visto en el filtro de Gauss y el operador *DOG*.

La convolución es fácil de implementar, tal como se mostró antes, pero el costo computacional es bastante alto, especialmente en el caso de una convolución bidimensional. Sin embargo, el filtro gaussiano es separable, es decir, como se vió antes, puede ser separado en dos convoluciones con Gaussianas unidimensionales, y la derivación puede ser obtenida después.

Luego se hallan las posiciones donde el gradiente es un máximo local a lo largo de la dirección del gradiente y se realiza una umbralización con el fin de encontrar los bordes cuyo gradiente es superior a un valor de umbral. Ver figura 5.8.



Figura 5.8: Imagen filtrada con el operador de Canny con $\sigma = 0.5$ y valor de umbral 100.

Versiones discretas de los operadores de gran tamaño definidos, como el de Canny en el dominio continuo, pueden ser realizados muestreando la función continua sobre una ventana de tamaño $W \times W$. El tamaño de la ventana debe ser lo suficientemente grande para evitar que el truncamiento de la función cause artefactos de alta frecuencia [41].

5.2.3 Aproximación de Deriche

Una extensión del filtro de banda limitada de Canny es presentada por Deriche. Así, él utiliza los mismos criterios de Canny, buscando la realización de un filtro de respuesta impulsional infinita (RII) [13]. El encuentra entonces, la misma ecuación diferencial para el filtro y solo modifica las condiciones en los límites al hacer $W \rightarrow \infty$:

$$h(0) = 0 ; h(\infty) = 0 ; h'(0) = s ; h'(\infty) = 0$$

cuya solución es:

$$h(x) = -c \cdot e^{-\alpha|x|} \cdot \operatorname{sen} \omega x$$

siendo α , ω y c reales positivos.

Evaluando para este operador las diferentes integrales que intervienen en el cálculo de los criterios de funcionamiento, se obtiene:

$$\lambda = \sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{2\alpha}{\alpha^2 + \omega^2}}$$

luego:

$$\Sigma.\lambda = \frac{2\alpha}{\sqrt{\alpha^2 + \omega^2}} \text{ y } r = \sqrt{\frac{\alpha^2 + \omega^2}{5\alpha^2 + \omega^4 + 6\alpha^2\omega^2}}$$

Haciendo $m = \frac{\alpha}{\omega}$, se presentan cuatro casos:

- a. Si $m \gg 1$, es decir $\alpha \gg \omega$: $\lambda = \sqrt{2\alpha}$ $\Sigma \approx \sqrt{\frac{2}{\alpha}}$ $\Sigma.\lambda \approx 2k \approx 0.44$
- b. Si $m \ll 1$ $\lambda = \sqrt{2\alpha}$ $\Sigma \approx \frac{\lambda}{m}$ $\Sigma.\lambda \approx 2m$ $k \approx 1$
- c. Si $m = 1$ $\lambda = \sqrt{2\alpha}$ $\Sigma \approx \sqrt{\frac{1}{\alpha}}$ $\Sigma.\lambda \approx 1.414$ $k \approx 0.58$
- d. Si $m = \sqrt{3}$ $\lambda = \sqrt{2\alpha}$ $\Sigma \approx \sqrt{\frac{3}{2\alpha}}$ $\Sigma.\lambda \approx 1.732$ $k \approx 0.5$

El caso d muestra que para el mismo valor de k , el funcionamiento del operador de Deriche es superior en más del 90% al de la primera derivada de una función gaussiana ($\Sigma.\lambda = 0.92$).

El caso c muestra que para un valor identico de r el filtro de Deriche h es mejor en más del 25% que el filtro óptimo obtenido por Canny con $\Sigma.\lambda \approx 1.12$.

El caso b presenta una respuesta ideal para k , pero el problema es que el producto $\Sigma.\lambda$ se hace mucho menor que 1.

El caso a muestra el mejor compromiso. Corresponde al límite del operador cuando $\omega \rightarrow 0$. Dado que $\Sigma.\lambda$ es máximo en $\omega = 0$ y que si $\omega.x$ es muy pequeño, se tiene que $\sin(\omega x) \approx \omega.x$. Este filtro óptimo puede escribirse como:

$$h(x) = -c.x.e^{-\alpha|x|}$$

De hecho, este operador es la solución de la ecuación diferencial 5.2 cuando el discriminante de su ecuación característica es cero.

c es una constante de normalización, permitiendo tener una respuesta máxima igual a uno para $x = 0$ dada una entrada escalón $u(x)$:

$$\int_{-\infty}^0 c.t.e^{-\alpha|t|}dt = c \int_{-\infty}^0 t.e^{-\alpha t}dt = 1 \implies c = -\alpha^2 \quad (5.3)$$

Para evitar el truncamiento del filtro, Deriche emplea una implementación recursiva de segundo orden. La aplicación de una formulación recursiva permite acelerar el filtrado y conseguir una mejor aproximación a la forma del filtro comparada con la técnica de filtrado mediante máscaras de convolución, la cual aproxima su forma mediante un muestreo truncado. Los resultados obtenidos con el filtro de Deriche muestran una gran mejora con respecto a los de Canny, en particular en cuanto hace a la reducción de respuestas múltiples [5].

A. Filtro de Suavización (Integración) de Deriche

El filtro de suavización de Deriche es un filtro de segundo orden definido por:

$$h(x) = k(\alpha|x| + 1)e^{-\alpha|x|}$$

donde k es escogido de manera que se obtiene un filtro discreto normalizado:

$$\sum_{-\infty}^{+\infty} h(n) = 1 \iff k = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}$$

y puede ser implementado recursivamente. Siendo $p[i, j]$ la imagen original y $q_x[i, j]$ la imagen filtrada luego de hacer el filtraje en la dirección x , se tiene:

$$q_1[i, j] = a_0 p[i, j] + a_1 p[i, j - 1] - b_1 q_1[i, j - 1] - b_2 q_1[i, j - 2] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

$$q_2[i, j] = a_2 p[i, j + 1] + a_3 p[i, j + 2] - b_1 q_2[i, j + 1] - b_2 q_2[i, j + 2] \quad \text{para } j = N, \dots, 1 \text{ e } i = 1, \dots, M$$

$$q_x[i, j] = q_1[i, j] + q_2[i, j] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

Para el montaje de los filtros recursivos se acostumbra hacer cero la entrada x y la salida q fuera de los límites de la imagen. Se tiene entonces:

$$p[i, 0] = q_1[i, 0] = q_1[i, -1] = 0$$

$$p[i, N + 1] = p[i, N + 2] = q_2[i, N + 1] = q_2[i, N + 2] = 0$$

Los coeficientes de las ecuaciones son respectivamente:

$$a_o = k$$

$$a_1 = k(\alpha - 1)e^{-\alpha}$$

$$a_2 = k(\alpha + 1)e^{-\alpha}$$

$$a_3 = -ke^{-2\alpha}$$

$$b_1 = -2e^{-\alpha}$$

$$b_2 = e^{-2\alpha}$$

Luego se realiza el mismo filtraje en la dirección y sobre $q_x[i, j]$ para obtener la imagen filtrada $q[i, j]$:

$$q_1[i, j] = a_0 p[i, j] + a_1 p[i - 1, j] - b_1 q_1[i - 1, j] - b_2 q_1[i - 2, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

$$q_2[i, j] = a_2 p[i + 1, j] + a_3 p[i + 2, j] - b_1 q_2[i + 1, j] - b_2 q_2[i + 2, j] \quad \text{para } i = M, \dots, 1 \text{ y } j = 1, \dots, N$$

$$q[i, j] = q_1[i, j] + q_2[i, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

donde:

$$p[0, j] = q_1[0, j] = q_1[-1, j] = 0$$

$$p[M + 1, j] = p[M + 2, j] = q_2[M + 1, j] = q_2[M + 2, j] = 0$$

La figura 5.9 presenta el resultado de aplicar el filtro de suavización sobre la imagen de referencia.



Figura 5.9: Imagen filtrada con el operador pasabajos de Deriche: a. $\alpha = 0.5$. b. $\alpha = 2$.

B. Filtro de primera derivación de Deriche

El filtro de la primera derivación de Deriche es un filtro de segundo orden definido por (ver figura 5.10):

$$h'(x) = -cxe^{-\alpha|x|}$$

implementado recursivamente de la siguiente manera:

$$q_1[i, j] = ap[i, j - 1] - b_1 q_1[i, j - 1] - b_2 q_1[i, j - 2] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

$$q_2[i, j] = -ap[i, j + 1] - b_1 q_2[i, j + 1] - b_2 q_2[i, j + 2] \quad \text{para } j = N, \dots, 1 \text{ e } i = 1, \dots, M$$

$$q_x[i, j] = q_1[i, j] + q_2[i, j] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

donde:

$$p[i, 0] = q_1[i, 0] = q_1[i, -1] = 0$$

$$p[i, N + 1] = q_2[i, N + 1] = q_2[i, N + 2] = 0$$

Luego se realiza el mismo filtraje en la dirección y sobre $q_x[i, j]$ para obtener la imagen filtrada $q[i, j]$:

$$q_1[i, j] = ap[i - 1, j] - b_1 q_1[i - 1, j] - b_2 q_1[i - 2, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

$$q_2[i, j] = -ap[i + 1, j] - b_1 q_2[i + 1, j] - b_2 q_2[i + 2, j] \quad \text{para } i = M, \dots, 1 \text{ y } j = 1, \dots, N$$

$$q[i, j] = q_1[i, j] + q_2[i, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

donde:

$$p[0, j] = q_1[0, j] = q_1[-1, j] = 0$$

$$p[M+1, j] = q_2[M+1, j] = q_2[M+2, j] = 0$$

Los coeficientes de las ecuaciones son respectivamente:

$$a = ce^{-\alpha}$$

$$b_1 = -2e^{-\alpha}$$

$$b_2 = e^{-2\alpha}$$

$$c = \frac{(1 - e^{-\alpha})^2}{e^{-\alpha}}$$

Puede notarse que la constante de normalización c es diferente de la calculada para un filtro continuo dada en 5.3.

Mediante la utilización de este filtro puede obtenerse el gradiente de la imagen.

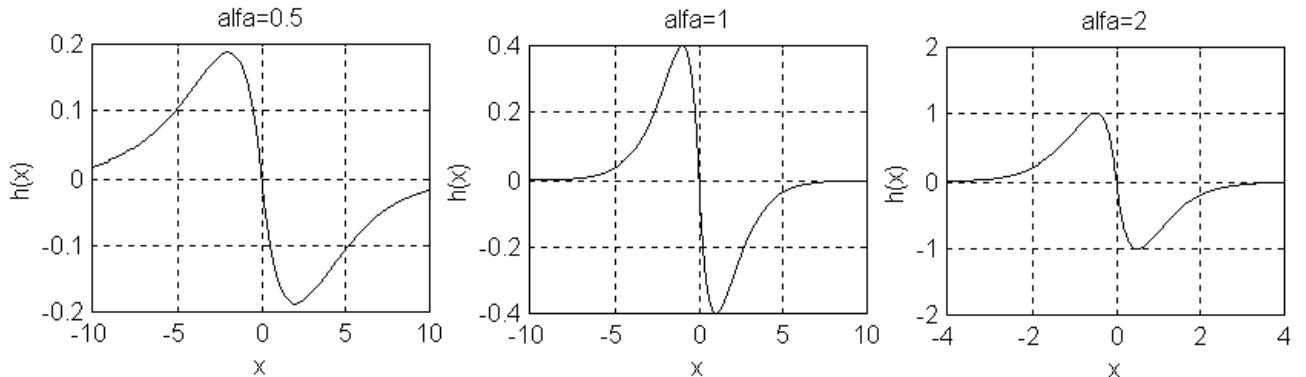


Figura 5.10: Forma del filtro de derivación de Deriche para $\alpha = 0.5, 1$ y 2 .

La figura 5.11 presenta el resultado de aplicar el filtro de derivación sobre las imágenes de la figura 5.9.

C. Filtro de segunda derivación de Deriche

El filtro de derivación de Deriche es un filtro de segundo orden definido por:

$$h''(x) = -(1 - k\alpha|x|) e^{-\alpha|x|}$$

donde d es la constante de normalización, es decir, escogida de manera que la respuesta a una señal constante sea cero:

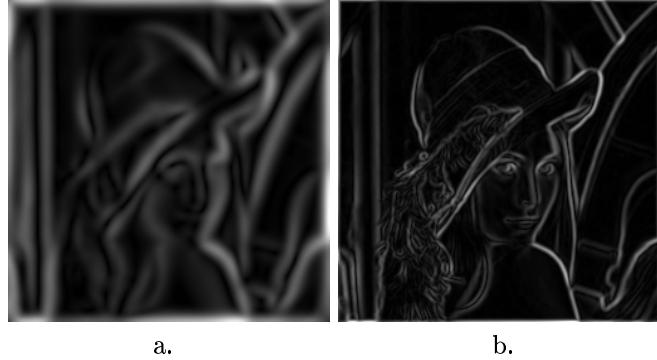


Figura 5.11: Imagenes resultantes al aplicar el operador de gradiente de Deriche a las imágenes de la figura 5.9:
a. $\alpha = 0.5$. b. $\alpha = 2$.

$$\sum_{-\infty}^{+\infty} h''(n) = 0 \iff k = \frac{1 - e^{-2\alpha}}{2\alpha e^{-\alpha}}$$

y es implementado recursivamente de la siguiente manera:

$$q_1[i, j] = -p[i, j] + a_1 p[i, j - 1] - b_1 q_1[i, j - 1] - b_2 q_1[i, j - 2] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

$$q_2[i, j] = a_2 p[i, j + 1] + a_3 p[i, j + 2] - b_1 q_2[i, j + 1] - b_2 q_2[i, j + 2] \quad \text{para } j = N, \dots, 1 \text{ e } i = 1, \dots, M$$

$$q_x[i, j] = q_1[i, j] + q_2[i, j] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

donde:

$$p[i, 0] = q_1[i, 0] = q_1[i, -1] = 0$$

$$p[i, N + 1] = p[i, N + 2] = q_2[i, N + 1] = q_2[i, N + 2] = 0$$

Luego se realiza el mismo filtraje en la dirección y sobre $q_x[i, j]$ para obtener la imagen filtrada $q[i, j]$:

$$q_1[i, j] = -p[i, j] + a_1 p[i - 1, j] - b_1 q_1[i - 1, j] - b_2 q_1[i - 2, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

$$q_2[i, j] = a_2 p[i + 1, j] + a_3 p[i + 2, j] - b_1 q_2[i + 1, j] - b_2 q_2[i + 2, j] \quad \text{para } i = M, \dots, 1 \text{ y } j = 1, \dots, N$$

$$q[i, j] = q_1[i, j] + q_2[i, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

donde:

$$p[0, j] = q_1[0, j] = q_1[-1, j] = 0$$

$$p[M+1, j] = p[M+2, j] = q_2[M+1, j] = q_2[M+2, j] = 0$$

Los coeficientes de las ecuaciones son respectivamente:

$$a_1 = (1 + k\alpha) e^{-\alpha}$$

$$a_2 = -(1 - k\alpha) e^{-\alpha}$$

$$a_3 = e^{-2\alpha}$$

$$b_1 = -2e^{-\alpha}$$

$$b_2 = e^{-2\alpha}$$

Mediante la utilización de este filtro puede obtenerse el laplaciano de la imagen. La figura 5.12 presenta el resultado de aplicar el filtro de derivación sobre las imágenes de la figura 5.9.

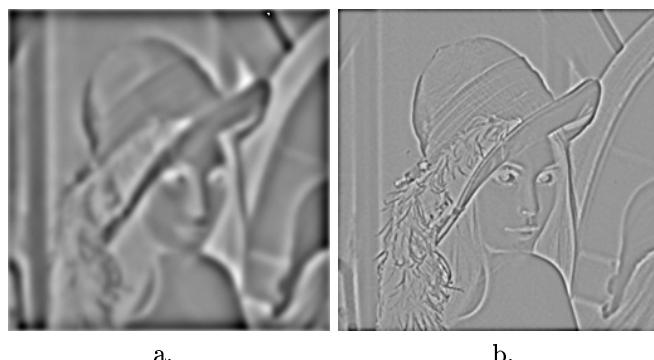


Figura 5.12: Imágenes resultantes al aplicar el operador de gradiente de Deriche a las imágenes de la figura 5.9:
a. $\alpha = 0.5$. b. $\alpha = 2$.

5.2.4 Aproximación de Shen y Castan

Este filtro filtro fue introducido por Shen y Castan para aproximar el laplaciano mediante la diferencia entre la imagen original y la imagen suavizada. Shen y Castan adoptan la forma de Deriche para la detección de bordes, es decir, primero se aplica sobre la imagen una suavización, realizada con un filtro pasabajos, seguida por la aplicación de un operador diferencial para realizar la búsqueda de pixeles correspondientes a los bordes. El criterio escogido busca una solución optima para los dos primeros criterios de Canny.

Ellos determinan primero las características de un filtro óptimo de suavización. Luego, utilizando las funciones de transferencia de este filtro y su primera y segunda derivadas, proponen varias soluciones para la detección de contornos.

Así, las consideraciones tomadas por Shen y Castan llevan a escoger una función $h(x)$ par y a analizar el rendimiento del filtro dentro de una ventana de tamaño $2W$. Ellos encuentran el siguiente criterio a minimizar:

$$C_n^2 = \frac{4 \int_0^W h^2(x) dx \int_0^W h'^2(x) dx}{h^4(0)}$$

Es decir, la función $h(x)$ continua que minimiza C_n es el filtro de suavización óptimo para el detector de bordes. La minimización de C_n^2 equivale a minimizar:

$$\Psi(x, h, h') = h'^2 + \lambda h^2$$

Para ello, se emplea la ecuación de Euler de primer orden:

$$\Psi_h - \frac{\partial \Psi'_h}{\partial x} = 0$$

donde:

$$\Psi_h = \frac{\partial \Psi}{\partial h}$$

y se llega a la ecuación diferencial:

$$2h''(x) - 2\lambda h(x) = 0$$

cuya solución es de la forma:

$$h(x) = b_1 e^{\alpha x} + b_2 e^{-\alpha x}$$

Teniendo en cuenta las condiciones en los límites $W \rightarrow \infty$, $\lim_{x \rightarrow \infty} h(x) = 0$ y dado que $h(x)$ es una función par, ellos encuentran que la función óptima es el filtro exponencial simétrico infinito:

$$h(x) = b e^{-\alpha|x|}$$

donde b es una constante de normalización definida por:

$$\int_{-\infty}^{\infty} h(\tau) d\tau = 1$$

de donde:

$$b = \frac{\alpha}{2} \implies h(x) = \frac{\alpha}{2} e^{-\alpha|x|}$$

En dos dimensiones, este filtro continuo está dado por:

$$h(x, y) = c e^{-\alpha(|x|+|y|)}$$

Discretizando la función continua $h(x)$ y escogiendo c de manera que se obtenga un filtro normalizado, es decir:

$$\sum_{-\infty}^{+\infty} h(n) = 1$$

se encuentra:

$$c = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}}$$

luego:

$$h[i, j] = \frac{(1 - b) b^{|x|+|y|}}{1 + b}$$

El filtro de derivación correspondiente $h'(x)$ normalizando la primera derivada del filtro de suavización:

$$h'(x) = \begin{cases} de^{-\alpha|x|} & \text{si } x \geq 0 \\ -de^{-\alpha|x|} & \text{si } x \leq 0 \end{cases}$$

d se escoge de manera que se obtenga un filtro normalizado en el espacio discreto:

$$\sum_{-\infty}^{+\infty} h'(n) = 1$$

de donde:

$$d = 1 - e^{-\alpha}$$

El parámetro α define el ancho del filtro; a medida que α es más pequeño la suavización efectuada por el filtro es más fuerte.

Este filtro puede ser implementado en la misma forma en que se implementa el filtro derivada de una función Gaussiana, aplicando sobre la imagen un filtro unidimensional primero en la dirección x y luego en la dirección y . Sin embargo, Shen y Castan también realizaron una implementación recursiva de su filtro. Siendo $p[i, j]$ la imagen original y $q_x[i, j]$ la imagen filtrada luego de hacer el filtraje en la dirección x , se tiene:

$$q_1[i, j] = \frac{1 - b}{1 + b} p[i, j] + b q_1[i, j - 1] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

$$q_2[i, j] = b \frac{1 - b}{1 + b} p[i, j] + b q_1[i, j + 1] \quad \text{para } j = N, \dots, 1 \text{ e } i = 1, \dots, M$$

$$q_x[i, j] = q_1[i, j] + q_2[i, j + 1] \quad \text{para } j = 1, \dots, N \text{ e } i = 1, \dots, M$$

donde:

$$p[i, 0] = 0$$

$$q_1[i, 0] = 0$$

$$q_2 [i, M + 1] = 0$$

Luego se realiza el mismo filtraje en la dirección y sobre $q_x [i, j]$ para obtener la imagen filtrada $q [i, j]$:

$$q_1 [i, j] = \frac{1 - b}{1 + b} p [i, j] + b q_1 [i - 1, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

$$q_2 [i, j] = b \frac{1 - b}{1 + b} p [i, j] + b q_1 [i + 1, j] \quad \text{para } i = M, \dots, 1 \text{ y } j = 1, \dots, N$$

$$q [i, j] = q_1 [i, j] + q_2 [i + 1, j] \quad \text{para } i = 1, \dots, M \text{ y } j = 1, \dots, N$$

donde:

$$p [0, j] = 0$$

$$q_1 [0, j] = 0$$

$$q_2 [N + 1, j] = 0$$

El parámetro b es fijado por el usuario donde $0 \leq b \leq 1$.

El filtro de derivación $h(x)$ corresponde a un valor óptimo del producto $\Sigma\Lambda$, es decir, para el mejor compromiso detección-localización. Shen y Castan sostienen que este filtro da una mejor relación señal a ruido con respecto al de Canny y provee una mejor localización. Esto puede deberse a que el filtro de Canny es tan solo una aproximación de la derivada de una función Gaussiana a su modelo, mientras que Shen y Castan emplean su operador óptimo o también podría deberse a una diferencia en la forma en que los diferentes criterios de optimización se ven reflejados en la práctica [38]. De otro lado, Shen y Castan no tienen en cuenta el criterio de unicidad de respuestas, por lo que es posible que este filtro produzca respuestas espurias en caso de ruido y bordes difusos. La discontinuidad de orden 1 en el punto $x = 0$ permite evitar una delocalización importante de los bordes en la imagen suavizada, incluso con valores pequeños de α . Sin embargo, esta discontinuidad puede implicar la detección de múltiples respuestas [34].

5.2.5 Anotaciones

Los mapas de bordes constituyen, en general, un paso intermedio para la ejecución de procesos posteriores. Puede ser, por ejemplo, empleada para el muestreo de puntos de un contorno con el fin de comprimir la información acerca de un objeto. Pueden buscarse formas geométricas básicas o primitivas, tales como elipses, rectas, círculos, en visión por computador o reconocimiento de formas, etc.

Capítulo 6

Selección de los píxeles del contorno

Aún cuando las técnicas del cálculo del gradiente y el laplaciano constituyen un paso importante en la etapa de detección de contornos o bordes en una imagen, no entregan directamente los píxeles del contorno. Por lo tanto, se debe introducir una etapa de selección de los píxeles del borde con el fin de obtener una imagen binarizada donde los píxeles que hagan parte del borde sean negros y blancos los demás.

6.1 Selección a partir del gradiente

Cuando se emplea un operador de gradiente para la detección de bordes, se dispone del gradiente en cada pixel de la imagen. La magnitud o norma del gradiente en el borde debe ser mayor que la magnitud de éste a cada lado del borde. De esta forma, la magnitud del gradiente en un pixel es grande si pertenece a un borde y es pequeña si no lo es. Varias técnicas han sido desarrolladas tomando en cuenta esta propiedad.

6.1.1 Umbralización simple

La técnica más simple y la primera en que podría pensarse consiste en seleccionar un valor de umbral de la magnitud del gradiente y tomar como valores del borde aquellos que sean superiores a él. La selección del valor de umbral puede hacerse estimando el número n de píxeles en la imagen que pertenecen al contorno y escogiendo como umbral aquel valor de la magnitud del gradiente que permita escoger como píxeles del contorno un número de píxeles cercano a n .

Sin embargo, aún cuando se haga una selección adecuada del valor de umbral, el contorno resultante puede resultar entrecortado. El rompimiento en el contorno obtenido se debe a las fluctuaciones del máximo gradiente por encima y por debajo del umbral escogido a lo largo del contorno, puesto que siempre habrá alguna fluctuación en los bordes de la imagen original debida al ruido, a pesar de éste sea bastante débil. Es de esperar, que solo alrededor de la mitad de los píxeles del borde estén por encima del umbral por lo que el borde obtenido no será totalmente continuo. Desafortunadamente este problema se presenta en la mayoría de casos y es muy difícil encontrar un valor de umbral bajo, es decir con una alta sensibilidad, y que a la vez no sea sensible a los falsos bordes producidos por ruido. Además, esta técnica entrega resultados bastante incorrectos cuando la magnitud del gradiente cambia fuertemente a lo largo de los píxeles del contorno, pudiendo ser superior al umbral en unas zonas e inferior en otras, seleccionando erroneamente como píxeles del contorno a otros cuya magnitud de gradiente también es alta. En realidad, no existe un valor de umbral que permita obtener correctamente los verdaderos píxeles del borde sin seleccionar erroneamente otros debido al ruido.

6.1.2 Umbralización por histeresis de los máximos locales del gradiente

John Canny propuso un método de umbralización adaptiva mediante el principio de histeresis con el fin de eliminar el rompimiento de los contornos de borde obtenidos con la técnica de umbralización simple.

Canny, busca mejorar la selección tomando como píxeles pertenecientes al borde sólo aquellos píxeles cuya magnitud del gradiente corresponde a un extremo local. Estos extremos locales corresponden a los pasajes por cero de la segunda derivada en la dirección del gradiente. En un paso posterior, se eliminan los píxeles

cuya magnitud del gradiente sea bastante baja y resulten eliminados aplicando la umbralización por histeresis. Esta técnica emplea la propiedad de conectividad entre los pixeles y permite obtener puntos de contorno bien conectados entre ellos [34].

A. Selección de los extremos locales del gradiente

A continuación se presentan dos técnicas para la selección de los extremos locales del gradiente. Asumiendo que el gradiente cambia de manera lineal y continua como función de la posición del pixel en la imagen, la magnitud del gradiente en cualquier punto puede ser estimada mediante la interpolación lineal de la magnitud del gradiente entre los pixeles vecinos al punto. Además, es bueno recordar que cada pixel en la imagen tiene una dirección de gradiente asociada.

Técnica general.

Sean p_1 y p_2 los dos puntos ubicados a una distancia d del centro de un pixel q , en la dirección del gradiente en q y en la dirección opuesta respectivamente (Ver figura 6.1). Las magnitudes del gradiente en p_1 y p_2 son calculadas mediante la interpolación lineal de las magnitudes del gradiente en sus pixeles vecinos. Por ejemplo, para el caso del punto p_1 se realiza una interpolación lineal de las magnitudes del gradiente en r , s , t y u , tomando en cuenta la distancia a p_1 .

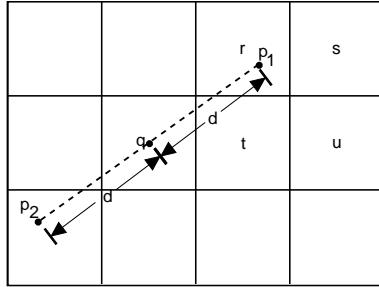


Figura 6.1: Selección de los extremos locales mediante la primera técnica.

El pixel q es seleccionado si $|\nabla q| > |\nabla p_1|$ y $|\nabla q| \geq |\nabla p_2|$. Al determinar como condición que el máximo sea estrictamente mayor en un sentido, hace que la localización del contorno se haga dentro de la zona de mayor o menor valor de los niveles de gris [34].

La figura 6.3 presenta el resultado de aplicar esta técnica sobre las imágenes de la figura 5.11.

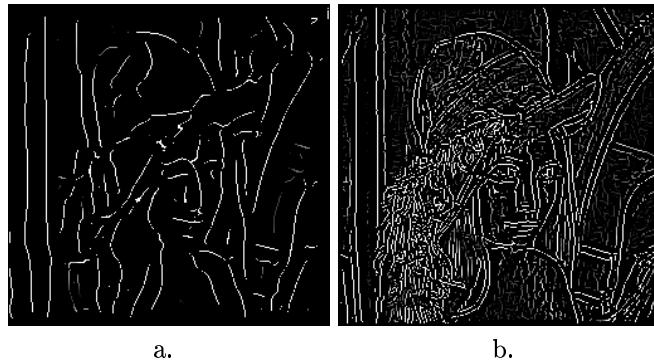


Figura 6.2: Imágenes resultantes al aplicar la primera técnica de detección de máximos locales para las imágenes procesadas con el operador de Deriche: a. $\alpha = 0.5$. b. $\alpha = 2$.

La segunda técnica toma en consideración que la distancia entre pixel y pixel, siguiendo los ejes x y y , es de una unidad. Suponiendo la existencia de líneas horizontales y verticales que conecten los pixeles vecinos al pixel de interés q , la magnitud del gradiente es calculada en el punto p donde la primera de estas líneas es atravesada moviéndose en la dirección del gradiente a partir de q e igualmente siguiendo la dirección opuesta. Se calcula

la magnitud del gradiente en el punto p , punto de cruce siguiendo la dirección del gradiente en q con la línea que une los pixeles vecinos r y t , mediante la interpolación de las magnitudes de los gradientes en r y t .

$$|\nabla p| = (r_y - p_y) |\nabla r| + (p_y - t_y) |\nabla t| = [1 - (p_y - t_y)] |\nabla r| + (p_y - t_y) |\nabla t| \quad (6.1)$$

donde los componentes x, y del gradiente en cada pixel son identificados por los subíndices.
Sea v la distancia entre p y t :

$$v = p_y - t_y \quad 0 \leq v \leq 1 \quad (6.2)$$

Reemplazando 6.2 en 6.1 se tiene:

$$|\nabla p| = (1 - v) |\nabla r| + v |\nabla t|$$

Con el fin de simplificar el cálculo de v pueden tomarse ocho casos, uno para cada octante del plano cartesiano. Se tiene que la dirección del gradiente en el pixel q está dada por:

$$\theta = \arctan\left(\frac{\nabla q_y}{\nabla q_x}\right)$$

Empleando la pendiente m de la recta por pasa p y q se tiene:

$$\theta = m = \frac{dy}{dx} = \frac{p_y - q_y}{p_x - q_x} = \frac{p_y - q_y}{1} = \frac{p_y - t_y}{1} = p_y - t_y = v$$

Luego:

$$v = \arctan\left(\frac{dy}{dx}\right)$$

B. Umbralización por histeresis de los extremos locales del gradiente

Esta técnica de umbralización, desarrollada por Canny, emplea dos valores de umbral para la detección del contorno: Uno alto t_h y uno bajo t_l . Inicialmente se definen como pertenecientes al borde todos aquellos pixeles cuya magnitud de gradiente sea superior a t_h . Luego se toman también como pixeles pertenecientes al borde, aquellos conectados a los ya definidos como pertenecientes al borde y cuya magnitud de gradiente es mayor que t_l . Algunas veces la traza de la secuencia de pixeles conectados al borde, cuya magnitud de gradiente es mayor que t_l y menor que t_h es limitada a un número específico de pixeles. t_h y t_l pueden ser determinados a partir del histograma acumulado de los valores de la magnitud del gradiente en los extremos locales [34].

Los resultados de la umbralización por histeresis pueden mejorar si luego de tomar los pixeles superiores a t_h como pertenecientes al borde, se realiza la búsqueda de los pixeles conectados a estos, no en todas las direcciones sino solamente en la dirección perpendicular a la dirección del gradiente, pues así se sigue la dirección del contorno. Monga afirma que en esta forma es posible disminuir el valor de umbral bajo t_l sin introducir contornos parasitos y es particularmente útil en la detección de contornos en tres dimensiones, en donde una expansión en todas las direcciones pueden introducirse falsos puntos en el contorno [34]. En esta técnica se aplican en forma implícita las restricciones sobre la morfología del contorno y no solamente la información proveniente de la función imagen [34].

La figura 6.3 presenta el resultado de aplicar esta técnica sobre las imágenes de la figura 6.2.

6.2 Selección a partir del laplaciano

Los puntos de los bordes están localizados en los pasajes por cero del laplaciano de la imagen $\Delta(q)$. Si el cálculo del laplaciano fuera exacto y no existiera ruido en la imagen bastaría con tomar los puntos donde el laplaciano fuera cero como puntos del contorno. Sin embargo, la aproximación del operador laplaciano introduce ruido. Por lo tanto, un procedimiento adicional es necesario.

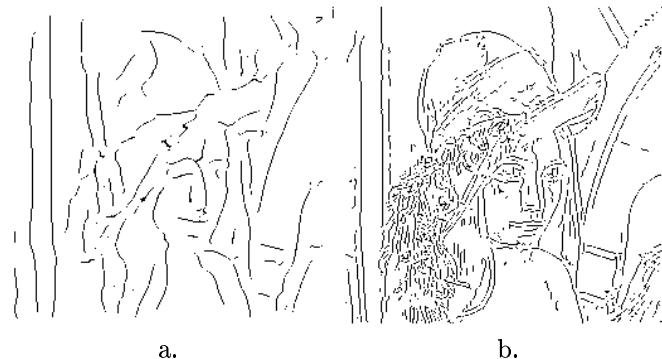


Figura 6.3: Imagenes resultantes al aplicar la técnica de histeresis con $t_l = 100$ y $t_h = 200$ a las imágenes de la figura 6.2: a. $\alpha = 0.5$. b. $\alpha = 2$.

6.2.1 Supresión de cruces por cero falsos

El método de supresión de los cruces por cero falsos es el correspondiente en el laplaciano a la técnica de retención de los máximos locales en el gradiente. Un cruce por cero del laplaciano corresponde a un máximo o un mínimo del gradiente. Si en el cruce por cero del laplaciano, ocurre un cambio de signo de positivo a negativo, éste se denominara positivo y será negativo en caso contrario. Sólo se consideran como pixeles pertenecientes a un borde aquellos situados en un cruce por cero positivo debido a un gradiente positivo, es decir un cambio entre un nivel de gris dado a otro más alto, y los cruces por cero negativos correspondientes a un gradiente negativo, los demás cruces son considerados erroneos.

Para evitar el cálculo del gradiente, se comparan los niveles de gris en la imagen original (normalmente ésta debe haber sido antes sometida a un filtraje pasabajo con el fin de eliminar ruido y cumplir la formulación correcta establecida por Poggio).

6.2.2 Umbralización por gradiente adaptivo

En algunas situaciones la técnica de supresión de cruces por cero falsos no es suficiente, especialmente cuando la imagen original es demasiado ruidosa. En este caso, la selección de pixeles puede ser afinada tomando como pixeles del borde sólo aquellos preseleccionados cuyo gradiente sea superior a un umbral.

Sin embargo, el cálculo adicional del gradiente hace esta técnica mucho más pesada computacionalmente. Puede emplearse en este caso, el método del gradiente adaptivo, sugerido por Shen y Castan para obtener una estimación de manera rápida. La técnica consiste en ubicar una ventana de lado l centrada sobre cada pixel donde se da un cruce por cero válido. Si el pixel corresponde a un borde, la ventana debe contener dos regiones de diferentes niveles de gris separadas por un borde, donde cada región puede ser identificada a partir de la imagen binarizada del laplaciano de la imagen obtenida por:

$$p(x, y) = \begin{cases} 1 & \text{si } \nabla q(x, y) > 0 \\ 0 & \text{si } \nabla q(x, y) \leq 0 \end{cases}$$

El gradiente estimado corresponde a la diferencia del nivel de gris entre las dos regiones. El último paso consiste en aplicar una umbralización escogiendo un valor de umbral del gradiente determinado o por el método de histeresis.

Capítulo 7

Segmentación por regiones

7.1 Generalidades

La umbralización, también denominada técnica de binarización, permite separar en regiones una imagen de acuerdo a un valor de gris óptimo seleccionado t^* basado en un criterio dado. Así, los píxeles por debajo del valor de umbral tomarán un valor de gris q_0 y los píxeles por encima del umbral el valor q_1 . Si t^* está determinado únicamente a partir del nivel de gris de cada pixel, se dice que el método es dependiente del punto. Si t^* se determina a partir de una propiedad local (por ejemplo: el nivel de distribución de gris local) en la vecindad de cada pixel, se dice que el método es dependiente de la región.

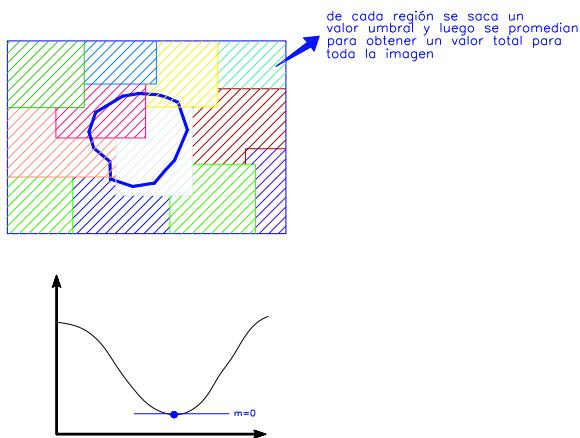


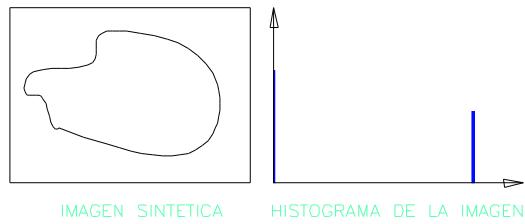
Figura 7.1: Técnica de segmentación por regiones por métodos locales.

Hay métodos de umbralización tanto globales como locales. En las técnicas globales se calcula un valor umbral único para toda la imagen, mientras que las técnicas locales partitionan la imagen en subimágenes y se determina un valor de umbral para cada una de ellas.

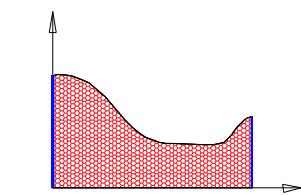
Cuando se tiene una imagen filtrada, con un pasaaltos como ya se ha estudiado, teóricamente se debe tener un histograma como se observa en la figura 7.2.

Así, para diferentes imágenes el histograma tiene diferentes formas, tales como las que se presentan en la figura 7.3. Observando la forma de cada histograma puede analizarse la distribución de la información registrada en la imagen de manera que es posible delimitar ciertas zonas. En algunos histogramas puede apreciarse una concentración de píxeles alrededor de unos picos de ciertos valores de intensidad separados entre sí por valles. Estas agrupaciones son denominadas modos del histograma.

Para llevar a cabo el proceso de binarización de una imagen es necesario buscar un valor de umbral $t \in L$, que permita aplicar una función de transformación definida como:



En la realidad el diagrama observado de la imagen tiene un comportamiento del siguiente tipo:



Este tipo de histograma se dice que tiene un comportamiento BINODAL.

Figura 7.2: Histograma de la imagen de referencia filtrada con un pasaaltos.

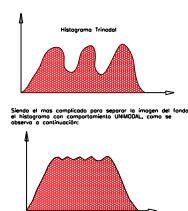


Figura 7.3: Diferentes formas de un histograma.

$$p(x, y) = \begin{cases} q_0 & \text{si } q(x, y) > t \\ q_1 & \text{si } q(x, y) \leq t \end{cases}$$

La figura 7.4 muestra la figura de ejemplo binarizada con un valor de umbral $t = 100$.



Figura 7.4: Imagen de referencia binarizada con $t = 100$.

Existen varios criterios para el cálculo de t^* . A continuación se presentan principalmente los métodos que se encuentran en el artículo de Sahoo et al. [46].

7.2 Métodos para el cálculo de un umbral óptimo

A continuación se presenta una breve síntesis de algunas técnicas de binarización, las cuales son utilizadas en el tratamiento de imágenes de acuerdo con distintos criterios según los requerimientos del caso. Se presentan técnicas de carácter global y local, así como los métodos que fueron utilizados para la implementación de algunas de ellas.

En algunos métodos es necesario conocer la probabilidad de ocurrencia de un nivel de gris en la imagen, la cual, como vimos antes está definida por $p(q_i) = \frac{n_i}{n}$ y es graficada en el histograma.

7.2.1 Binarización global

Estas técnicas, como se mencionó antes, sólo encuentran un único valor umbral y a partir de éste se binariza la imagen. Pueden ser divididas en dos grupos: técnicas dependientes del punto, en las cuales el umbral es determinado con base en el nivel de gris de cada pixel y dependientes de la región, en las que el umbral se determina a partir de una propiedad en la vecindad de cada pixel.

A. Técnicas dependientes del valor del punto

a. Método del promedio Este método es uno de los más simples, ya que se toman los diferentes niveles de gris en los que la imagen está definida, se suman y el promedio de esta suma se toma como el umbral óptimo t^* . La figura 7.5.b presenta el resultado de aplicar este método sobre la imagen de referencia 7.5.a.

b. Método p-mosaico En esta técnica se asume que la imagen a tratar consta de objetos bastante contrastados, por ejemplo objetos oscuros y fondo claro y se requiere conocer el porcentaje p en pixeles del área de la imagen que ocupa el objeto a segmentar, es decir, se debe conocer el tamaño en pixeles del objeto. El umbral óptimo t^* corresponde al nivel de gris que cubre por lo menos $(100 - p)\%$ de los pixeles dentro de los objetos en la imagen binarizada, así

$$t^* = t_{\min} \mid \frac{\# \text{Pixeles_objeto}}{\# \text{Pixeles_imagen}} * 100\% \geq (100 - p)\%$$

Por ejemplo, si un objeto cubre el 20% de una imagen, entonces la imagen será binarizada al nivel de gris que permita que por lo menos el 20% de los pixeles sean delineados dentro del objeto.

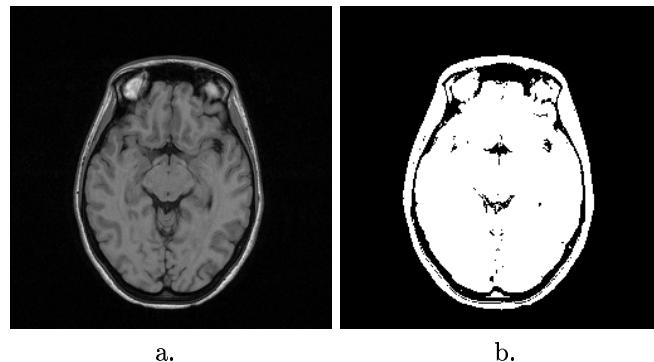


Figura 7.5: Técnicas de binarización. a. Imagen original. b. Imagen binarizada por el método del valor promedio.

c. Método del modo En imágenes donde es fácil distinguir el objeto de interés del fondo de la imagen, el histograma se caracteriza por ser bimodal, en tal caso, un valor de umbral puede ser escogido entre los valores de gris que se encuentran en el valle que separan los dos modos (picos) del histograma.

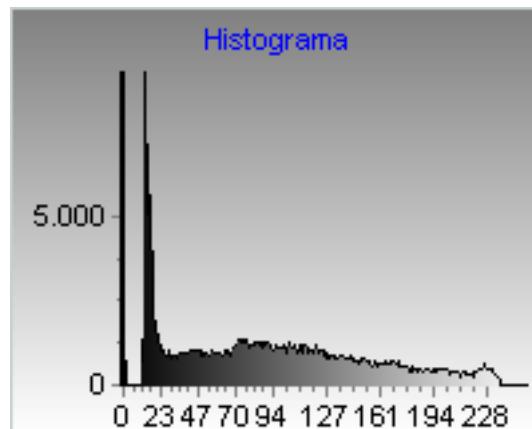


Figura 7.6: Histograma bimodal.

Por ejemplo, en el histograma de la figura 7.6 un posible valor de umbral estaría en 23 más o menos.

$$t^* = \{t \mid \min [h(g)]\}$$

d. Método de Ostu Este método se basa en el análisis del discriminante. Así, la operación de binarización es vista como la selección de un valor de umbral que partitiona los pixeles de la imagen en dos clases $C_0 = \{0, 1, 2, \dots, t\}$ y $C_1 = \{t + 1, t + 2, \dots, L - 1\}$ (por ejemplo, el fondo y el objeto de interés).

Un valor de umbral óptimo t^* puede ser encontrado minimizando la relación entre las covarianzas del objeto y el fondo, es decir, se busca que el valor de umbral separe dos clases donde cada pixel se acerque más al valor esperado de cada clase:

$$t^* = t \mid \min \left\{ \frac{\text{Covarianza}_1}{\text{Covarianza}_2} \right\}$$

y esta relación puede ser obtenida minimizando uno de los siguientes criterios con respecto a t :

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}$$

$$\eta = \frac{\sigma_x^2}{\sigma_T^2}$$

$$\kappa = \frac{\sigma_T^2}{\sigma_W^2}$$

donde:

- σ_x^2 es la varianza en cada clase: $\sigma_x^2 = \text{var}(x) = \text{cov}(xx) = E(x^2) - [E(x)]^2 = \mu(x^2) - \mu_x^2$
- σ_W^2 es la covarianza entre las clases C_0 y C_1 : $\sigma_W^2 = \text{cov}(C_0, C_1) = E(C_0, C_1) - E(C_0)E(C_1)$
- σ_T^2 es la varianza total.

El criterio más fácil de calcular es η y tenemos:

$$t^* = t \mid \min \eta$$

$$\omega\varpi_0 = \sum_{i=0}^t p_i$$

$$\varpi\omega_1 = \sum_{i=t+1}^{L-1} p_i = 1 - \omega_0$$

$$\mu_t = \sum_{i=0}^t ip_i$$

$$\mu_T = \sum_{i=0}^{L-1} ip_i$$

$$\mu_0 = \frac{\mu_t}{\omega_0}$$

$$\mu_1 = \frac{\mu_T - \mu_t}{1 - \omega_0} = \frac{\mu_T - \mu_t}{\omega_1}$$

$$\sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_T)^2 p_i$$

$$\sigma_B^2 = \omega\varpi_0\omega_1 (\mu_0\mu_1)^2$$

La figura 7.7 presenta el resultado de la binarización por el método de Ostu de la figura 7.5.a.

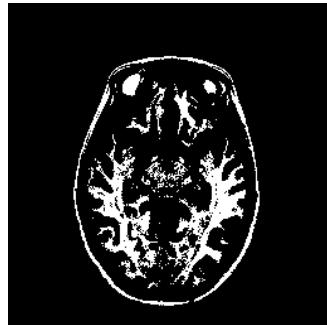


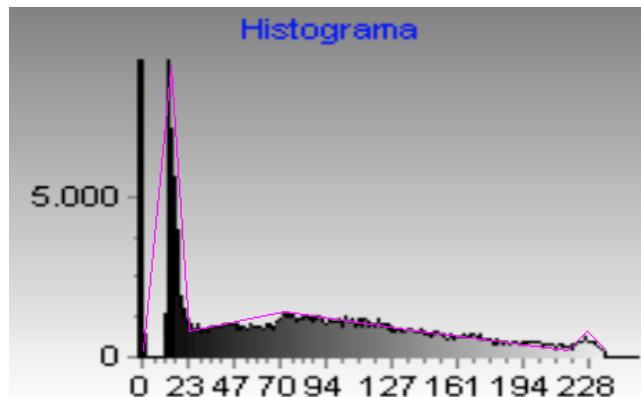
Figura 7.7: Imagen binarizada por el método de Ostu.

e. Método por análisis de la concavidad del histograma En las imágenes donde sobre el fondo hay varios objetos es posible seleccionar el valor de umbral usando el método del modo. Sin embargo, cuando no es relativamente fácil hallar los valles del histograma, entonces se puede hallar t^* en un pico del histograma.

Si H es el histograma definido sobre un conjunto de niveles de gris q_0, q_1, \dots, q_{L-1} , entonces la altura de dicho histograma para cada nivel de gris puede verse como:

$h(q_0), h(q_1), h(q_2), \dots, h(q_{L-1})$ donde $h(q_i) \neq 0 \forall i$. En esta forma el histograma H puede ser visto como una región bidimensional.

Para determinar las concavidades de H se construye el polígono convexo \overline{H} más pequeño que contenga a H , tal como se aprecia en la figura 7.8.

Figura 7.8: Polígono convexo \overline{H} del histograma de una imagen.

Las concavidades del histograma se determinan de la diferencia $H - \overline{H}$. Los posibles valores de umbral son aquellos para los cuales la diferencia $\overline{h}(q_i) - h(q_i)$ tiene un máximo local. Sin embargo, varios máximos locales espurios pueden ser introducidos debido a ruidos de tipo spike. Una medida de balance E_i ha sido presentada con el fin de eliminarlos:

$$E_i = \left\{ \sum_{j=q_0}^{q_{i-1}} h(j) \right\} \left\{ \sum_{j=q_i}^{q_{L-1}} h(j) \right\}$$

E_i mide el balance alrededor del nivel de gris q_i . Los valores espurios se presentan generalmente en los lados del histograma donde hay pocos pixeles con niveles de gris correspondientes a estas zonas y donde el ruido es, en consecuencia, más notable. Para los valores espurios E_i es pequeño y por tanto las concavidades espurias pueden ser descartadas.

f. Métodos basados en la entropía Estos métodos están basados en la teoría de la información de Claude Shannon [8]. Aquí la información, se refiere más al conocimiento que aporta la información; en el caso de imágenes, podremos decir que la información se refiere al conocimiento de qué niveles de gris pertenecen al fondo y cuáles al objeto a segmentar. El aspecto crucial de la teoría es la medida de la información: la entropía. Obviamente, la medida de la información está relacionada con la incertidumbre de saber cuáles pixeles pertenecen al objeto de interés y es evidente que incluye a las probabilidades. Así, el histograma es considerado como una fuente de información de L símbolos y el umbral óptimo se obtiene teniendo en cuenta el mayor grado de información (menor grado de desorden) que se posea. Por tanto, en estos métodos se toma como valor de umbral aquel en donde la entropía es mínima.

i. Método de Pun Pun ha desarrollado dos algoritmos para obtener el valor de umbral. Tomando t como valor de umbral, se pueden definir dos entropías a posteriori. La primera puede ser vista como la medida de la información a posteriori asociada con los pixeles por debajo del nivel de umbral y correspondiente a los pixeles negros luego de la binarización:

$$H'_n(t) = - \sum_{i=0}^t p_i \log_e p_i$$

La segunda corresponde a la medida de la información a posteriori asociada con los pixeles blancos:

$$H'_b(t) = - \sum_{i=t+1}^{L-1} p_i \log_e p_i$$

Conociendo la entropía a priori del histograma, Pun propuso un primer algoritmo para determinar el valor de umbral óptimo maximizando el límite superior de la entropía a posteriori:

$$H'(t) = H'_n(t) + H'_b(t)$$

$$t^* = t \mid \max H'(t)$$

Pun mostró que maximizar $H'(t)$ es equivalente a maximizar con respecto a t la función:

$$f(t) = \frac{H_t}{H_T} \frac{\log_e P_t}{\log_e \max\{p_0, \dots, p_t\}} + \left[1 - \frac{H_t}{H_T} \right] \frac{\log_e (1-P_t)}{\log_e \max\{p_{t+1}, \dots, p_{L-1}\}}$$

donde:

$$H_t = - \sum_{i=0}^t p_i \log_e p_i$$

$$H_T = - \sum_{i=0}^{L-1} p_i \log_e p_i$$

$$P_t = - \sum_{i=0}^t p_i$$

En un segundo algoritmo, Pun propone el uso de un coeficiente de anisotropía α :

$$\alpha = \frac{\sum_{i=0}^m p_i \log_e p_i}{\sum_{i=0}^{L-1} p_i \log_e p_i}$$

y m es el más pequeño entero tal que:

$$\sum_{i=0}^m p_i \geq 0.5$$

El valor de umbral óptimo t^* es escogido de manera que:

$$\sum_{i=0}^{t^*} p_i = \begin{cases} 1 - \alpha & \text{si } \alpha \leq 0.5 \\ \alpha & \text{si } \alpha > 0.5 \end{cases}$$

Sin embargo, ha sido demostrado que este algoritmo siempre produce un valor de umbral $t^* \geq m$, introduciendo un sesgo innecesario [26].

ii. Método de Kapur, Sahoo y Wong En éste método, a partir de la distribución de probabilidad original de los niveles de gris de la imagen, son obtenidas dos distribuciones. La distribución de probabilidad del fondo:

$$\frac{p_0}{P_t}, \frac{p_1}{P_t}, \dots, \frac{p_t}{P_t}$$

y la distribución de probabilidad del objeto de interés:

$$\frac{p_{t+1}}{1 - P_t}, \frac{p_{t+2}}{1 - P_t}, \dots, \frac{p_{L-1}}{1 - P_t}$$

donde t es el valor de umbral y $P_t = \sum_{i=0}^t p_i$.

La entropía de los pixeles negros está dada por:

$$H_n(t) = - \sum_{i=0}^t \frac{p_i}{P_t} \log_e \left(\frac{p_i}{P_t} \right)$$

y la medida de información de los pixeles blancos por:

$$H_b(t) = - \sum_{i=t+1}^{L-1} \frac{p_i}{1 - P_t} \log_e \left(\frac{p_i}{1 - P_t} \right)$$

El valor de umbral óptimo t^* es definido como el nivel de umbral que maximiza $H_n(t) + H_b(t)$:

$$t^* = t \mid \max \{H_n(t) + H_b(t)\}$$

iii. Método de Johansen y Bille Este método divide el conjunto de niveles de gris en dos partes con el fin de minimizar la independencia, en el sentido de la teoría de la información, entre ellas. El umbral óptimo se encuentra:

$$t^* = t \mid \min \{S(t) + \bar{S}(t)\}$$

donde:

$$S(t) = \log_e \left(\sum_{i=0}^t p_i \right) - \frac{1}{\sum_{i=0}^t p_i} \left[p_t \log_e p_t + \left(\sum_{i=0}^{t-1} p_i \right) \log_e \left(\sum_{i=0}^{t-1} p_i \right) \right]$$

$$\overline{S}(t) = \log_e \left(\sum_{i=t}^{L-1} p_i \right) - \frac{1}{\sum_{i=t}^{L-1} p_i} \left[p_t \log_e p_t + \left(\sum_{i=t+1}^{L-1} p_i \right) \log_e \left(\sum_{i=t+1}^{L-1} p_i \right) \right]$$

iv. Método de preservación del momento En este método, los valores de umbral son hallados determinísticamente, de tal forma que los momentos de la imagen a ser binarizada son preservados en la imagen binarizada. El i -ésimo momento está dado por:

$$m_i = \frac{1}{n} \sum_{q=0}^{L-1} q^i h(q) \quad i = 1, 2, 3, \dots$$

donde n es el número total de pixeles de la imagen.

El umbral óptimo t^* es obtenido a partir del histograma, escogiendo t^* como la p_0 -ésima pieza, donde:

$$p_0 = \frac{z - m_1}{(c_1^2 - 4c_0)^{\frac{1}{2}}}$$

$$c_0 = \frac{m_1 m_3 - m_2^2}{m_2 - m_1^2}$$

$$c_1 = \frac{m_1 m_2 - m_3}{m_2 - m_1^2}$$

$$z = \frac{1}{2} \left\{ (c_1^2 - 4c_0)^{\frac{1}{2}} - c_1 \right\}$$

v. Método del error mínimo En este método, el histograma es visto como un estimado de la función de densidad de probabilidad $p(q)$ de la población mezclada, comprendida tanto por los niveles de gris del objeto como por los pixeles de fondo. En general, se asume que cada uno de los dos componentes $p(q|i)$ de la mezcla tienen una distribución normal con media μ_i , desviación estándar σ_i y una probabilidad a priori P_i :

$$p(q) = \sum_{i=1}^2 P_i p(q|i)$$

donde:

$$p(q|i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(q-\mu_i)^2}{2\sigma_i^2}}$$

El valor de umbral puede ser obtenido resolviendo la ecuación:

$$\frac{(q - \mu_1)^2}{\sigma_1^2} + \log_e \sigma_1^2 - 2 \log_e P_1 = \frac{(q - \mu_2)^2}{\sigma_2^2} + \log_e \sigma_2^2 - 2 \log_e P_2$$

sin embargo, los valores de la media μ_i , la varianza σ_i^2 y la probabilidad P_i ($i = 1, 2$) de la densidad mezclada $p(q)$ asociada con la imagen a ser binarizada no son usualmente conocidos. Entonces se utiliza el siguiente criterio:

$$J(t) = 1 + 2 \{P_1(t) \log_e \sigma_1(t) + P_2(t) \log_e \sigma_2(t)\} - 2 \{P_1(t) \log_e P_1(t) + P_2(t) \log_e P_2(t)\}$$

donde:

$$P_1(t) = \sum_{q=0}^t h(q)$$

$$P_1(t) = \sum_{q=t+1}^{L-1} h(q)$$

$$\mu_1(t) = \frac{\left\{ \sum_{q=0}^t h(q) q \right\}}{P_1(t)}$$

$$\mu_2(t) = \frac{\left\{ \sum_{q=t+1}^{L-1} h(q) q \right\}}{P_2(t)}$$

$$\sigma_1^2(t) = \frac{\left\{ \sum_{q=0}^t h(q) [q - \mu_1(t)]^2 \right\}}{P_1(t)}$$

$$\sigma_2^2(t) = \frac{\left\{ \sum_{q=t+1}^{L-1} h(q) [q - \mu_2(t)]^2 \right\}}{P_2(t)}$$

y el umbral óptimo corresponde a:

$$t^* = t \mid \min J(t)$$

B. Técnicas dependientes de la región

a. Métodos de transformación del histograma En estos métodos no se escoge directamente un valor de umbral. En lugar de ello, el histograma es transformado con miras a obtener uno, donde los valles y picos sean más pronunciados y donde el método del modo visto antes pueda ser utilizado para hallar el umbral. El nuevo histograma es obtenido como resultado de una ponderación de los pixeles de acuerdo a propiedades locales de éstos. Los métodos asumen que cada imagen consta del fondo y objetos de interés, los cuales tienen una distribución unimodal de los niveles de gris.

i. Métodos basados en las estadísticas de segundo orden de los niveles de gris Una de las desventajas de los métodos de binarización dependientes del punto, es que ellos dependen únicamente de las estadísticas de primer orden de los niveles de gris de la imagen, como por ejemplo el histograma. Los métodos a continuación están basados en las estadísticas de segundo orden.

a) Método de matriz de co-ocurrencia La matriz de co-ocurrencia $M(d, \theta)$ introducida por Haralick para el análisis de texturas, es una matriz donde sus entradas son la frecuencia relativa de hallar dos píxeles vecinos cuyos niveles de gris sean i y j , separados una distancia d en una orientación θ .

M puede ser definida como:

$$M = M_{(1,0)} + M_{(1,\frac{\pi}{2})} + M_{(1,\pi)} + M_{(1,\frac{3\pi}{2})}$$

es decir, que el elemento (i, j) de M corresponde a la frecuencia con que el nivel de gris i ocurre como un vecino en conectividad 4 de un nivel de gris j .

M puede usarse para definir dos nuevos histogramas:

Un histograma basado en las entradas cercanas a la diagonal de M , en donde el histograma debería tener un profundo valle entre el fondo y el objeto.

El segundo histograma es construido, basado en las entradas fuera de la diagonal de M , en el cual debería tenerse un pico puntudo entre los niveles de gris del fondo y aquellos del objeto.

7.3 Método de crecimiento por regiones

En el método de segmentación por crecimiento de regiones, un conjunto inicial de pequeñas áreas son fusionadas iterativamente de acuerdo a ciertas condiciones de similaridad entre ellas.

Inicialmente se selecciona arbitrariamente un pixel que sirve de semilla y se compara con los píxeles vecinos (ver figura 7.9). La región a la cual pertenece el pixel semilla comienza a crecer, adicionando los píxeles vecinos que le son similares, incrementando de esta forma el tamaño de la región. Este proceso se repite hasta encontrar todos los píxeles pertenecientes a la región escogida. Cuando el crecimiento de una región termina, se escoge un nuevo pixel como semilla entre los que no pertenecen aún a ninguna región y se recomienza de nuevo el proceso. El proceso termina completamente cuando cada pixel de la imagen pertenece a alguna región. Esta técnica permite usualmente obtener muy buenas segmentaciones que corresponden bien a los bordes observados. El criterio de comparación más empleado es la similaridad en el valor de nivel de gris entre un pixel y su vecino.

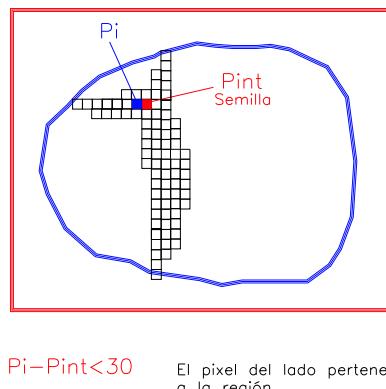


Figura 7.9: Método de crecimiento por regiones.

Sin embargo, el hecho de comenzar a segmentar a partir de un único pixel semilla y dejar crecer esta región completamente antes que otras, sesga la segmentación en favor de las que crecen primero. Esto puede producir efectos indeseables, tales como, en primer lugar, dada la región que domine el proceso de crecimiento, puede que ciertas ambigüedades alrededor de los bordes de las regiones adyacentes no sean resueltas correctamente y en segundo lugar, la escogencia de semillas diferentes puede producir resultados diferentes en la segmentación. Un problema adicional se puede presentar en el caso en que el pixel semilla arbitrariamente escogido caiga justamente en un borde.

Para tratar de resolver estos problemas, diversos métodos que permiten el crecimiento simultáneo de las regiones han sido desarrollados, así:

- Las similaridades de las regiones vecinas entre sí, son tomadas en cuenta en el proceso de crecimiento.
- No se permite que ninguna región domine completamente el procedimiento de crecimiento.
- Varias regiones de la imagen crecen simultáneamente. Lo que puede llevar a que regiones similares vecinas puedan gradualmente unirse.

Estos métodos pueden ser implementados fácil y eficientemente mediante técnicas de procesamiento paralelo [33].

7.4 Método de dividir y mezclar (splitting & merging)

La técnica de dividir y mezclar, conocida en inglés como splitting and merging, tiene un enfoque opuesto al de crecimiento por regiones. Basado en los métodos de divide y conquistarás, la idea básica es subdividir la imagen en un conjunto de regiones disyuntas y éstas a su vezirlas subdividiendo hasta encontrar que cada subregión es homogénea según algún criterio de similaridad dado. Luego entremezclar las regiones vecinas con características iguales [33][20].

Un caso particular de subdivisión, se da cuando se divide cada región en cuatro cuadrantes, dando origen a la técnica conocida como quadtree y que puede ser resumida en el siguiente algoritmo (Ver figura 7.10):

1. Inicialmente tomar toda la imagen como zona a analizar.
2. Revisar las zonas a estudiar y verificar si estas cumplen un criterio de similaridad especificado. Si una zona cumple el criterio, corresponde a una región de la imagen; sino dividirla en 4 cuadrantes disyuntos y tomarlos como nuevas zonas a analizar.
3. Mezclar dos regiones adyacentes, si el criterio sigue siendo válido para la unión de ellas.
4. Parar cuando no sean posibles más mezclas o subdivisiones.

En el peor de los casos, el tamaño de una región será de un pixel.

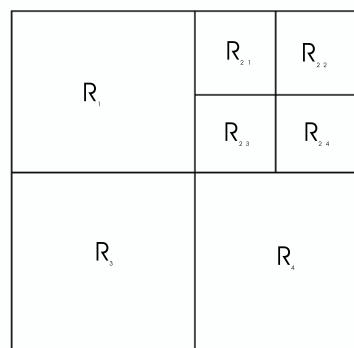


Figura 7.10: Método subdivisión y mezcla en quadtree.

El proceso de subdivisión de la imagen puede ser descrito por una estructura de árbol y como se observa en la figura 7.11, cada rama no terminal del árbol tendrá al menos cuatro subramas, a menos que un proceso de mezcla entre éstas haya tenido lugar.

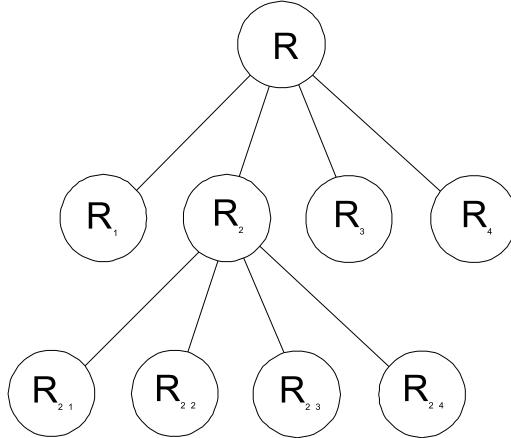


Figura 7.11: Quadtree correspondiente a la figura 7.10.

7.5 Etiquetado

El etiquetado de una imagen binaria es una operación que asigna un valor único, conocido como rótulo o etiqueta, a los pixeles que pertenecen a una misma región conectada. Dependiendo de la definición de conectividad, diferentes resultados pueden ser obtenidos:

La conectividad 4 para unir los pixeles correspondientes a los objetos de la imagen y la conectividad 8 para los pixeles correspondientes al fondo.

La conectividad 8 para unir los pixeles correspondientes a los objetos de la imagen y la conectividad 4 para los pixeles correspondientes al fondo.

La figura 7.12 ilustra un ejemplo donde se encuentran dos regiones aisladas tanto en conectividad 4, como en conectividad 8.

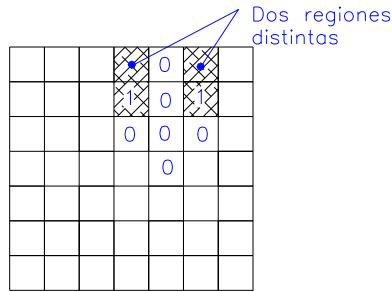


Figura 7.12: Etiquetado.

Existen varias técnicas para realizar el etiquetado. Una técnica sencilla presentada en la figura 7.13 requiere dos barridos de la imagen y una tabla de etiquetas. La imagen se recorre de izquierda a derecha y de arriba hacia abajo. Se revisa cada pixel q cuyo valor sea uno y según el caso de conectividad aplicado se analizan sus pixeles vecinos, ya verificados, con el fin de constatar si q se encuentra conectado a alguno de ellos. Tomando el caso de conectividad 4, se revisa si el vecino arriba de q tiene también un valor de uno; de ser así, se asigna a q la misma etiqueta. Luego se revisa el pixel a la izquierda de q ; si también tiene un valor de uno, se verifica si su etiqueta es igual a la de q . De no serlo se asigna en la tabla una equivalencia entre estas dos etiquetas. En caso de que q no esté conectado a ningún pixel arriba ni a su izquierda, se le asigna una nueva etiqueta y se inscribe en la tabla. Luego se revisa la tabla y cada etiqueta que tenga una equivalente se le reasigna la etiqueta de menor valor. En un segundo barrido de la imagen, se revisa cada pixel etiquetado y se revisa la tabla, en caso de tener una etiqueta equivalente, se le asigna ésta.

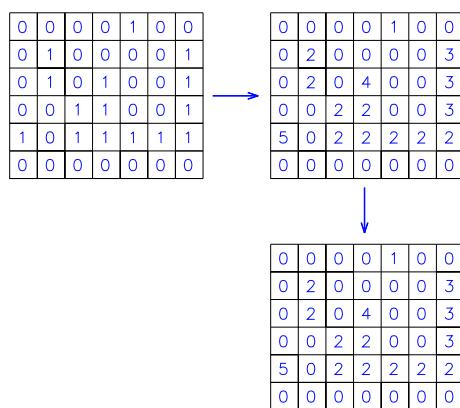


Figura 7.13: Ejemplo numérico de conectividad.

Capítulo 8

Segmentación por multiresolución

En las técnicas de segmentación por multiresolución se busca segmentar la imagen en regiones de interés, intentando obtener una vista global de ella examinándola en diferentes niveles de resolución. La resolución más baja provee una visión global de la imagen y la más alta permite ver el detalle.

En el acercamiento basado en la generación de una pila, se crea una estructura de árbol para describir la imagen, definiendo una relación entre los pixeles en imágenes suavizadas sucesivamente de la imagen original. Así, se asocia cada pixel en una imagen con una intensidad máxima local. Este acercamiento se enfoca en la descomposición de una imagen en zonas claras y oscuras; cada una de ellas contenida en otras exceptuando la región representando la imagen completa [32].

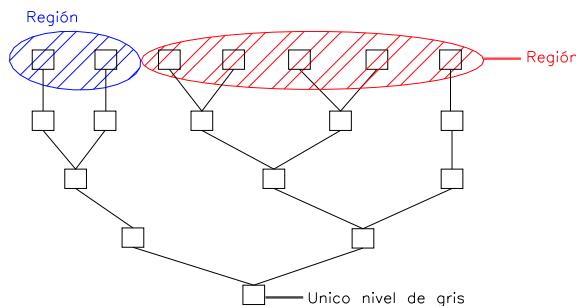


Figura 8.1: Método de segmentación por multiresolución.

Capítulo 9

Detección de primitivas

Las formas geométricas básicas que aproximan los bordes de los objetos en una imagen se conocen en la computación gráfica como primitivas, éstas incluyen los puntos, las líneas rectas y curvas, etc. En el análisis de una imagen resulta muy útil ser capaz de identificar los bordes que siguen las formas de unas primitiva específicas, tales como líneas rectas, círculos, elipses o curvas paramétricas que en ella puedan encontrarse. Así, puede ser posible la identificación de objetos según su forma.

9.1 Transformada de Hough

Una técnica muy empleada en la detección de curvas simples o primitivas en imágenes y en visión por computador es la transformada de Hough. Para comprender la forma en que esta transformada permite la detección de una primitiva, resulta más sencillo analizar primero el caso de la detección de la curva más simple, la línea recta, para luego extenderla a otras primitivas más complejas.

9.1.1 Transformada de Hough de líneas rectas

En este caso, la técnica se basa en la transformación del sistema de coordenadas cartesiano en otro cuyos ejes corresponden a la pendiente y al término independiente de la ecuación de la recta.

Todas las rectas que pasan por un pixel en la imagen ubicado en la posición (x_q, y_q) puede ser representadas por la ecuación:

$$y_q = mx_q + b \quad (9.1)$$

para distintos valores de m y b , donde m es la pendiente de cada recta y b su punto de intersección con el eje y . Ver figura 9.1.a.

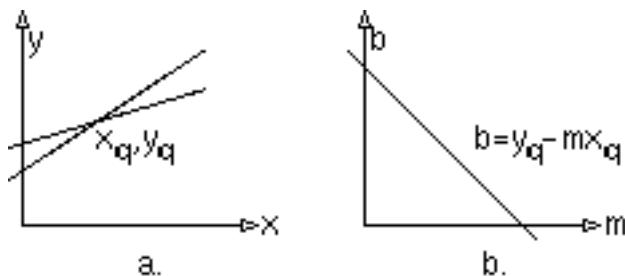


Figura 9.1: Representación de las rectas que pasan por el punto (x_q, y_q) : a. En el plano x,y . b. En el plano m,b .

La ecuación 9.1 puede ser reescrita como:

$$b = y_q - mx_q \quad (9.2)$$

donde ahora x_q y y_q son consideradas constantes y, m y b variables.

Al hacer la ecuación de la recta dependiente de la inclinación o pendiente m y no de las coordenadas de los puntos por las que pasa, y dada la discretización del espacio en las imágenes a tratar, se encuentra un sistema de coordenadas en el cual cada línea es representada por un solo punto con coordenadas (m_q, b_q) . De esta manera, la figura 9.1.b presenta la gráfica de m vs. b , donde cada línea que pasa por el punto (x_q, y_q) en el espacio (x,y) corresponde a cada uno de los puntos que forman la línea en el espacio (m,b) .

Considerando ahora dos pixeles p y q ubicados sobre una misma línea en el espacio (x,y) , se tiene:

- Todas las líneas que posiblemente pasan por un pixel pueden ser representadas en el espacio (m,b) por una única línea recta.
- Una línea simple que pasa a través de p y q en el espacio (x,y) aparece en el espacio (m,b) como la intersección de dos líneas correspondientes a p y q , tal como se ilustra en la figura 9.2.

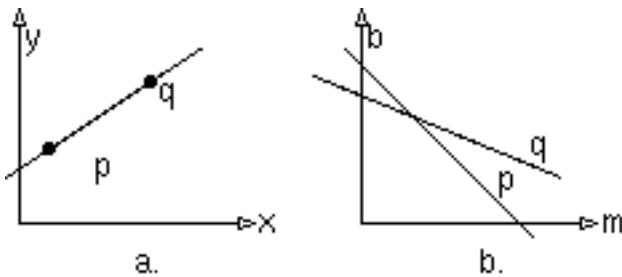


Figura 9.2: Correspondencia entre: a. Una línea que pasa por los puntos p y q en el espacio (x,y) . b. La intersección de las rectas p y q en el espacio (m,b) .

Generalizando se tiene:

- Todos los pixeles ubicados sobre una misma línea en el espacio (x,y) aparecen representados por líneas que pasan a través de un único punto en el espacio (m,b) .
- El único punto a través del cual las líneas pasan entrega los valores de m y b en la ecuación de la línea $y = mx + b$.

Con el fin de detectar las líneas rectas en una imagen, se debe cuantizar el espacio (m,b) en un arreglo bidimensional $\mathbf{H}(m,b)$ con un número adecuado de casillas en m y b , siendo ahora m el conjunto discreto de posibles pendientes y b un número finito de términos independientes, tal como se ilustra en la figura 9.3. Cada casilla dentro del arreglo \mathbf{H} es una celda acumuladora que contiene el número de pixeles que cumplen la ecuación de la recta correspondiente a los valores de m y b de esa casilla.

El proceso de detección de las rectas en la imagen, se realiza entonces de la siguiente forma:

Se inicializa el arreglo $\mathbf{H}(m,b)$ en cero. Luego se calculan los valores de b , según la posición (x_q, y_q) de cada pixel que cae sobre un borde en la imagen original de acuerdo a la ecuación $b = y_q - mx_q$ para $m_{\min} \leq m \leq m_{\max}$, incrementando en uno el valor de cada celda acumuladora de $\mathbf{H}(m,b)$ correspondiente.

Finalmente se buscan las casillas en el arreglo $\mathbf{H}(m,b)$ que contengan los valores más elevados. Cada casilla corresponde a una línea en la imagen original.

Una línea recta en una imagen necesariamente tiene dos puntos terminales p y q , sin embargo, la transformada de Hough entrega una línea infinita para cada par de valores de m y b , en lugar del segmento de línea definido entre los puntos p y q . Gracias a esta propiedad de la transformada, los pixeles que conforman la línea no deben ser necesariamente contiguos. Esta propiedad es útil en la detección de líneas entrecortadas o con algunas aberturas debidas a ruido o cuando son ocultadas parcialmente por otros objetos. Sin embargo, cuando los bordes rectos de diferentes objetos estén alineados puede conducir a un resultados erróneo, pues serán identificados por una única línea [33].

Para el cálculo de la transformada de Hough, la representación implícita de la línea recta $y = mx + b$ no es práctica, pues a medida que las líneas tienden a hacerse paralelas al eje y la pendiente m se hace extremadamente grande, tiendiendo a infinito, y no resulta posible construir el arreglo $\mathbf{H}(m,b)$.

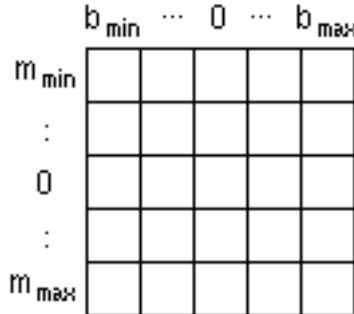


Figura 9.3: Arreglo acumulador bidimensional $\mathbf{H}(m,b)$ empleado en el algoritmo de Hough.

Para evitar este problema, es mejor emplear una representación paramétrica de la línea recta, la forma polar:

$$x \cos \theta + y \sin \theta = r$$

donde x y y son consideradas constantes y, θ y r variables.

A diferencia de la representación en el espacio (m,b) , al hacer la ecuación de la recta dependiente del ángulo y el radio y no de las coordenadas de los puntos por los que pasa, al considerar dos pixeles p y q ubicados sobre una misma línea en el espacio (x,y) , se tiene:

- Todas las líneas que posiblemente pasan por un pixel pueden ser representadas en el espacio (m,b) por una única línea sinusoida.
- Una línea simple que pasa a través de p y q en el espacio (x,y) aparece en el espacio (θ, r) como la intersección de dos sinusoides correspondientes a p y q .

Se construye así, un arreglo discreto $H(\theta, r)$ inicializado en cero y se calculan los valores de r , según la posición (x_q, y_q) de cada pixel que cae sobre un borde en la imagen original de acuerdo a la ecuación $x_q \cos \theta + y_q \sin \theta = r$ para $\theta_{min} \leq \theta \leq \theta_{max}$, incrementando en uno el valor de cada celda acumuladora de $H(\theta, r)$ correspondiente. El rango de r varía entre $\pm \sqrt{\text{cols}^2 + \text{filas}^2}$, donde cols y filas corresponden al número de columnas y filas de la imagen. θ puede variar entre $\pm 90^\circ$ o entre 0° y 180° con respecto al eje x .

La figura presenta la línea principal hallada en la imagen aplicando la transformada de Hough.

9.1.2 Transformada de Hough de otras primitivas

La transformada de Hough puede ser empleada para detectar otras formas de bordes en una imagen, especialmente aquellas formas geométricas que pueden ser descritas por pocos parámetros. Por ejemplo, para encontrar círculos se requieren tres parámetros: las coordenadas de su centro a y b , y su radio r . Así, se parte de la ecuación implícita del círculo:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (9.3)$$

Donde cada punto en el espacio (x,y) corresponde a una superficie en el espacio (a,b,r) en el cual dos de las variables pueden cambiar, r y a ó b , y la tercera se determina a partir de la ecuación 9.3. Por lo tanto, el método aplicado para la línea recta debe modificarse con el fin de emplear un arreglo acumulador tridimensional $\mathbf{H}(a,b,r)$, donde la casilla correspondiente a cada pixel de un borde que satisface la ecuación del círculo es incrementada en uno [33]. Si se conoce de antemano el valor del radio, solo será necesario emplear un arreglo bidimensional.

Una forma más complicada como la elipse requiere de un espacio de cinco parámetros [49]. Así, se parte de la ecuación implícita de la elipse:

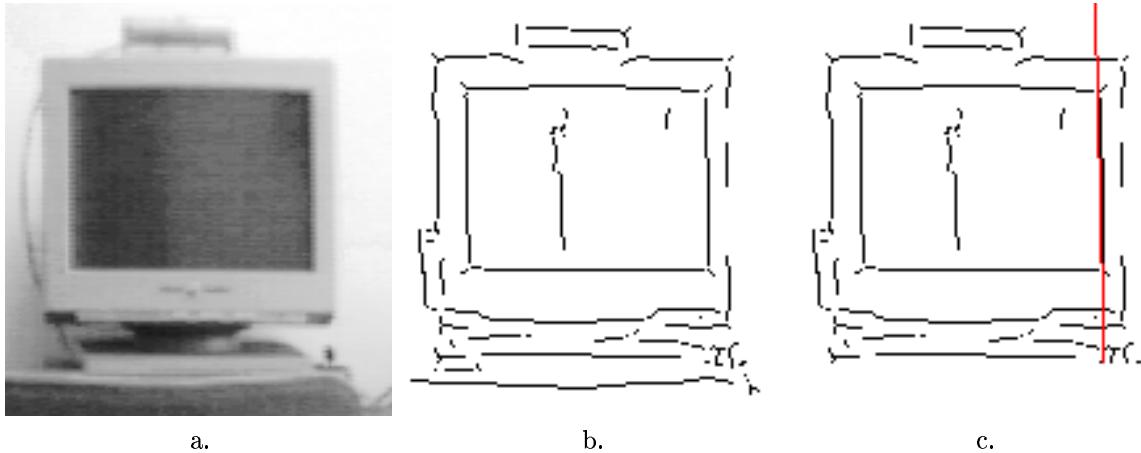


Figura 9.4: Ejemplo de aplicación de la transformada de Hough: a.Imagen original. b. Imagen tratada con filtro pasabajos y gradiente de Deriche $\alpha = 1$, detección de máximos locales e histeresis $t_l = 100$ y $t_h = 200$. c. Línea hallada con la transformada de Hough.

$$\left(\frac{x-a}{r_x}\right)^2 + \left(\frac{y-b}{r_y}\right)^2 = r^2$$

siendo a y b su centro, y r_x y r_y su radio mayor y menor respectivamente.

Esta técnica puede ser generalizada para detectar cualquier forma que pueda ser representada por un número limitado de segmentos de línea, la cual puede aparecer con cualquier orientación, posición y tamaño o escala. En este caso, el espacio de parámetros a utilizar, es en general de 4 dimensiones, dos para la posición, uno para la orientación y otro para su escala. Es útil en este caso extraer de antemano las líneas rectas[49].

Un estudio del ruido que puede ser introducido en las celdas acumuladoras y el sesgo inherente ha sido realizado por Brown [6].

Capítulo 10

Descriptores de región

El área de una región de la imagen se define como el número de píxeles contenidos en su contorno. El perímetro corresponde a la longitud de su contorno. Estas dos medidas se utilizan a veces como descriptores de la región.

Sea una población de vectores aleatorios \mathbf{X} de dimensión d de la forma:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

El vector medio de esta población se define como:

$$\mathbf{m}_x = E(\mathbf{X})$$

donde $E(\mathbf{X})$ es el valor esperado de la población de vectores \mathbf{X} . El cálculo del valor esperado de un vector se realiza evaluando los valores esperados de cada componente del vector.

La matriz de covarianza de la población de vectores \mathbf{X} es definida como:

$$\mathbf{C}_x = E \left\{ (\mathbf{X} - \mathbf{m}_x) (\mathbf{X} - \mathbf{m}_x)^T \right\}$$

en donde T indica el valor transpuesto. Dado que \mathbf{X} es de dimensión d , la matriz de covarianza \mathbf{C}_x es de orden $d \times d$. El elemento c_{ii} corresponde a la varianza del i -ésimo elemento x_i de la población de vectores \mathbf{X} y el elemento c_{ij} a la covarianza entre los elementos x_i y x_j de la población.

\mathbf{C}_x tiene varias propiedades útiles, es real y simétrica. Además, si sus elementos x_i y x_j no están correlacionados, su covarianza es cero, es decir, $c_{ij} = c_{ji} = 0$.

Cuando se tienen n vectores de muestra de una población aleatoria, el vector medio puede ser aproximado por:

$$\mathbf{m}_x = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$$

y la matriz de covarianza por:

$$\mathbf{C}_x = \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \mathbf{X}_k^T - \mathbf{m}_x \mathbf{m}_x^T$$

Si los pixeles interiores de una región son utilizados como variables aleatorias de este tipo, los vectores propios de la matriz de covarianza resultante se denominan ejes principales de la región. Los dos vectores propios apuntan en la dirección de máxima expansión de la región y están sujetos a la restricción de ser ortogonales. Otras medidas utilizadas como descriptores de región son la media, la mediana y los valores máximo y mínimo de los niveles de gris y el número de pixeles con valores por encima y por debajo de la media.

Un método importante aplicado en la descripción de regiones consiste en cuantificar su contenido de textura. No existe una definición formal de la textura. Sin embargo, este descriptor entrega unas medidas intuitivas acerca de algunas propiedades de las regiones en la imagen tales como su suavidad, rugosidad y regularidad. Existen básicamente tres métodos empleados para describir las texturas de una región: estadísticos, estructurales y espectrales. Las soluciones estadísticas proporcionan información acerca de características tales como la suavidad, rugosidad, granularidad y otras similares. Las técnicas estructurales tratan acerca de la composición de elementos primitivos de la imagen y las técnicas espectrales están asociadas con el análisis en el espectro de Fourier.

Uno de los métodos estadísticos más simples consiste en construir el histograma de la imagen y extraer los momentos de éste. Sin embargo, este método no aporta información referente a la posición de un pixel con respecto a sus vecinos. Para realizar esta clase de análisis, se recurre a la construcción de una matriz de co-ocurrencia de la imagen.

Sea P un operador de posición y \mathbf{A} una matriz de tamaño $k \times k$, cuyo elemento $a_{i,j}$ corresponde al número de veces que aparecen pixeles con nivel de gris q_i en la posición especificada por P en relación con los pixeles cuyo nivel de gris es q_j , donde $0 < i, j < k$. De esta forma, el tamaño de A está determinado por el número de niveles de gris diferentes en la imagen. Si n es el número total de pares de puntos que satisfacen el operador P , es posible dividir cada elemento de \mathbf{A} entre n para dar origen a matriz \mathbf{C} , de tal manera que cada valor $c_{i,j}$ resultante es un indicador probabilístico. Dado que \mathbf{C} depende directamente de P , se puede detectar la presencia de un patrón de textura dado eligiendo un operador de posición apropiado. Diversos descriptores pueden ser obtenidos a partir de P . Sin embargo, la complejidad del algoritmo para el cálculo de la matriz de co-ocurrencia resulta ser en la mayoría de los casos demasiado elevada.

Capítulo 11

Modelos de contorno activo: Snakes

11.1 Introducción

El reconocimiento de objetos en una imagen, por medio de la detección de rasgos o formas particulares, es uno de los problemas centrales que abarca el procesamiento de imágenes. Con tal objetivo se han desarrollado múltiples técnicas de seguimiento de bordes y contornos, sin poder llegar a decirse que exista una que sea mejor que las demás, debido a que cada técnica presenta un desempeño diferente dependiendo del entorno (características de la imagen) en que se utilice. En esta sección se presenta la definición y funcionamiento de uno de estos procedimientos: los modelos de contorno activo o snakes.

A grandes rasgos un snake tiene un comportamiento similar a una banda de caucho que se suelta sobre un objeto; de esta manera, al encogerse tiende a fijarse sobre el borde de éste y de esta forma mostrar su contorno. El snake puede ser definido como un modelo deformable con capacidad de movimiento, que obedece a la influencia de tres energías: interna, de la imagen y externa o restrictiva. Con una correcta combinación en la minimización de estas energías se orienta el snake hacia los rasgos distintivos de la imagen [27].

11.2 Modelos deformables

Los modelos deformables son una clase de primitivas creadas para el modelado de curvas, superficies y sólidos, diseñadas principalmente para facilitar la creación de animaciones por computador con alto grado de realismo [45]. Estos modelos pueden emular una gran variedad de comportamientos naturales como la elasticidad, la viscoelasticidad, la fractura y el derretimiento entre otros.

11.2.1 Clasificación

Dependiendo de las propiedades físicas que intenten emular, los modelos deformables se clasifican en:

A. Modelos elásticos

Este tipo de modelos representan materiales elásticos, por lo tanto retornan a su posición original por medio de una energía restauradora que es proporcional a la magnitud de la deformación.

B. Modelos inelásticos

Los modelos inelásticos emulan comportamientos mecánicos asociados comúnmente con sólidos altamente polimerizados como la plastilina, es decir, objetos que una vez deformados no retornan a su estado original [45].

C. Modelos termoelásticos

Este subtipo de modelos deformables puede representar características físicas como la conducción térmica, la termoelasticidad, el derretimiento y el comportamiento de fluidos.

Conociendo ahora el concepto de modelo deformable, se puede afirmar que los snakes son una aplicación de dichos modelos en el procesamiento de imágenes, que tiene como objetivo localizar bordes y contornos en éstas.

11.3 Estructura de los modelos de contorno activo

En esta sección, el modelo inicial o punto de partida del snake es definido por un conjunto de puntos de control unidos con líneas rectas como se observa en la figura 11.1. Bajo otras circunstancias es más conveniente utilizar arcos o líneas suavizadas para unir estos puntos. El modelo inicial puede ser cerrado (primer y último punto de control unidos) o abierto.

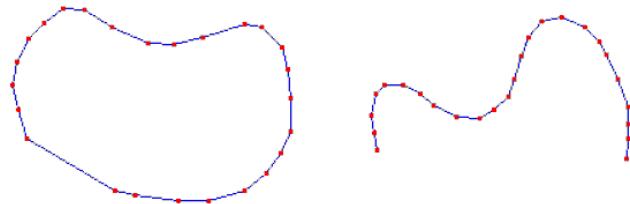


Figura 11.1: Modelo inicial cerrado y modelo inicial abierto. Los puntos rojos corresponden a los puntos de control.

Creado el modelo inicial, el siguiente paso es definir las energías que orientarán este modelo hasta los rasgos distintivos de la imagen sobre la cual se aplique. Como se mencionó antes, son tres las energías que pueden llegar a influenciar el movimiento del snake: la interna, la externa y la de la imagen:

$$E_{\text{snake}} = E_{\text{interna}} + E_{\text{imagen}} + E_{\text{externa}}$$

Para que un snake tenga movimiento no es necesario implementar las tres energías, con la interna o la de la imagen basta, pero seguramente no se obtendrán los resultados óptimos en cuanto a detección de rasgos se refiere.

Una observación conveniente es que el término energía se usa para establecer una analogía con los sistemas físicos, pero no debe tomarse en toda su dimensión, ya que en realidad se trata de pseudo-energías que cumplen con algunas características de las relacionadas con el concepto tradicional.

11.3.1 Energía interna

La energía interna determina el comportamiento del modelo de contorno activo, independientemente de las características de la imagen sobre la cual se aplique. Los comportamientos más utilizados son: el modelo de membrana o banda elástica, que hace que el snake se reduzca de tamaño y se ajuste a una forma ubicada en su interior. Si en el área encerrada por el modelo inicial se encuentran dos objetos bien definidos, el snake podría llegar a separarse y rodearlos. Otro comportamiento utilizado frecuentemente es el de una lámina delgada, también conocido como "tira de metal", el cual permite que el contorno del modelo cambie de forma pero sólo con curvaturas suaves y sin fuerza reductora.

Una descripción un poco más formal de la energía interna del modelo de contorno activo es [27]:

$$E_{\text{interna}} = \frac{a(s)|Vs(s)|^2 + b(s)|Vss(s)|^2}{2}$$

En donde el término de primer orden controlado por $a(s)$ hace que el snake se comporte como una membrana y el término de segundo orden controlado por $b(s)$ determina el comportamiento de lámina delgada. Variando los valores de los pesos $a(s)$ y $b(s)$ puede obtenerse un amplio rango de conductas para el modelo de contorno activo.

11.3.2 Energía elástica

Para explicar la energía elástica es posible realizar una analogía con el sistema físico de un resorte; entre mayor sea la longitud del resorte estirado, mayor será su energía potencial. Por tanto, si el modelo inicial posee puntos de control distantes, estos se orientarán rápidamente hacia el interior del snake [49].

Para crear una función de energía que describa el comportamiento de banda elástica en un modelo de contorno activo, es necesario imaginar que los puntos de control del modelo inicial se encuentran unidos por pequeños resortes, cada uno con una constante de elasticidad K_1 . Ahora es fácil determinar que la fuerza sobre un punto de control, está influenciada por el cuadrado de las distancias a cada uno de los puntos de control adyacentes. Así la energía interna puede ser escrita como [49]:

$$E_{elástica} = K_1 * \sum_{i=0}^n d_{i,i-1}^2$$

En donde:

n es el número de puntos de control.

$d_{i,i-1}$ es la distancia entre los puntos de control i y $i - 1$.

El cuadrado de la distancia entre dos puntos de control se determina utilizando el teorema de Pitágoras:

$$d_{i,i-1} = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2$$

Donde (x_i, y_i) son las coordenadas del punto de control i . El siguiente paso es transformar esta energía en movimiento, es decir, determinar la fuerza que se aplica sobre cada uno de los puntos de control; esta fuerza tiene un componente horizontal y otro vertical. Utilizando algunas herramientas de cálculo diferencial, se obtiene [49]:

$$F_{elástica_x,i} = 2 * K_1 * ((x_{i-1} - x_i) + (x_{i+1} - x_i))$$

$$F_{elástica_y,i} = 2 * K_1 * ((y_{i-1} - y_i) + (y_{i+1} - y_i))$$

Una representación de estas fuerzas es presentada en la figura 11.2.

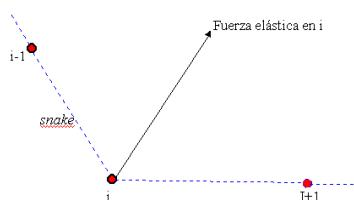


Figura 11.2: Representación geométrica de la fuerza elástica en el punto de control i .

Para obtener el movimiento del punto de control i debido a las componentes de la fuerza elástica, se crean las siguientes expresiones [49]:

$$x_i(t+1) = x_i(t) + [K_2 * F_{elástica_x,i}]$$

$$y_i(t+1) = y_i(t) + [K_2 * F_{elástica_y,i}]$$

Los índices $t + 1$ y t representan la nueva posición del snake y la posición inmediatamente anterior respectivamente. La constante K_2 determina la velocidad de desplazamiento del punto de control. En la figura 11.3 se puede observar un ejemplo del comportamiento de banda elástica en un snake cerrado.

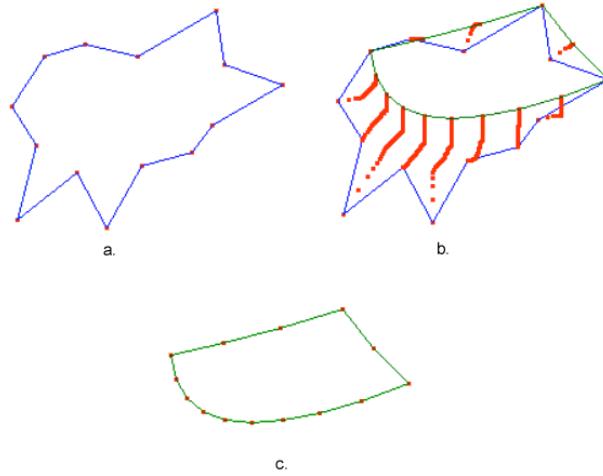


Figura 11.3: Ejemplo del comportamiento de banda elástica de un snake cerrado, con $K_1 = 2$, $K_2 = 0.05$ con una iteración de 100 pasos. a. Modelo inicial. b. Trayectoria de los puntos de control. c. Estado final del snake.

11.3.3 Energía de la imagen

El objetivo de definir la energía de la imagen es orientar el desplazamiento de los puntos de control del modelo de contorno activo hacia rasgos distintivos de la imagen. Estos rasgos son las líneas, los bordes y las terminaciones.

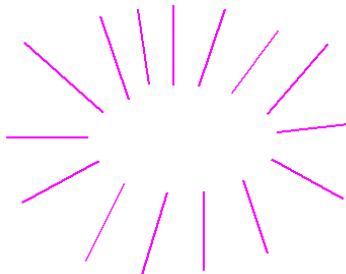


Figura 11.4: Contorno subjetivo de una elipse, creado a partir de terminaciones de líneas.

Las líneas determinan cambios de intensidad en la imagen, los bordes permiten identificar formas definidas y las terminaciones pueden permitir el establecimiento de contornos subjetivos, como el mostrado en el ejemplo de la figura 11.4. De esta forma, la energía de la imagen puede verse como [27]:

$$E_{imagen} = \omega_{linea}E_{linea} + \omega_{borde}E_{borde} + \omega_{term}E_{term}$$

Cada término de esta expresión tiene su respectivo peso ω que determina la importancia de cada una de las sub-energías.

A. Aporte energético de los bordes

La representación funcional de la energía aportada por los bordes es [27]:

$$E_{borde} = -|I(x, y)|^2$$

El gradiente de un vector indica la dirección de la máxima variación de una función en un punto; entonces el snake se orienta hacia los contornos que tienen altas variaciones de intensidad en la imagen; por lo tanto,

cualquiera de las técnicas vistas para la obtención del gradiente puede ser empleada para orientar el gradiente hacia el contorno del objeto.

B. Energía externa

Al implementar esta energía es posible definir puntos de atracción y puntos de repulsión sobre el área de la imagen y así influenciar el movimiento del snake acercando o alejando los puntos de control a las áreas definidas por el usuario.

11.4 Aplicaciones

Las aplicaciones de los snakes son amplias. Ya en el artículo original publicado por Kass et al. [27] se mencionan varias de sus posibles aplicaciones. Entre ellas se tienen la detección y seguimiento de contornos. En la figura 11.5 se ilustra un ejemplo en la detección del contorno de la cabeza en imágenes médicas. Los snakes han sido también empleados en el reconocimiento de rasgos faciales [12].

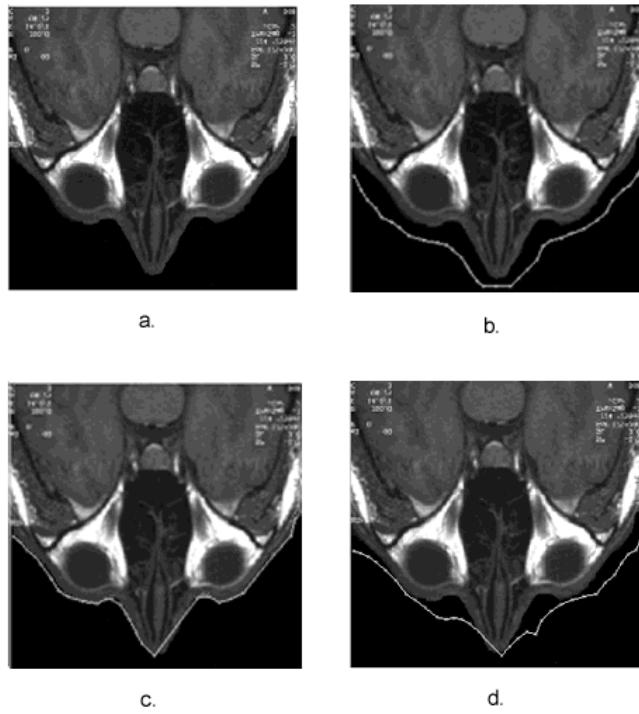


Figura 11.5: Ejemplo de aplicación de snakes. a. Imagen original. b. Modelo inicial o punto de partida del modelo de contorno activo. c. Ubicación del snake después de 100 pasos bajo la influencia de la energía de la imagen únicamente. d. Posición final del snake bajo la influencia de la energía de la imagen y una energía elástica con $K_1 = 1$ y $K_2 = 0.4$.

Los snakes son igualmente usados en la detección de bordes y contornos en secuencias de vídeo [27]. Un buen ejemplo de ello es una aplicación desarrollada en Finlandia que permite el conteo y seguimiento de objetos en una secuencia de imágenes [23]. Así mismo, los snakes son empleados en el ajuste de imágenes estereoscópicas [27] y su uso se ha extendido a la detección de bordes, contornos y superficies en imágenes en tres dimensiones [50][48], donde se les suele llamar con el nombre de globos (en inglés balloons).

Capítulo 12

Morfología matemática

La morfología matemática es una herramienta empleada para extraer componentes de la imagen que son útiles en la representación y en la descripción de la forma y la estructura de una región, tales como límites, esqueletos y el polígono convexo. La morfología matemática investiga esta estructura y la compara con distintas formas llamada elementos estructurantes. El lenguaje de la morfología matemática está basado en la teoría de conjuntos, en el uso de operadores (intersección, unión, inclusión, complemento) y en la adición y substracción de Minkowsky de subespacios en el espacio euclídeo empleados para transformar la imagen. Los operadores morfológicos no son filtros lineales. Usualmente la imagen transformada tiene pocos detalles, implicando una pérdida de información, pero sus características principales están aún presentes. Así de la imagen simplificada, diversos cálculos pueden ser realizados con el fin de hacer un análisis cuantitativo de la imagen.

También hay interés en las técnicas de pre o pos-procesamiento, tales como filtraje morfológico, adelgazamiento y pruning.

Los conjuntos en la morfología matemática representan los objetos. Para entender las operaciones morfológicas es conveniente definir la translación.

Sea A un conjunto en el n -espacio E^n , se define la translación de A por un elemento x como [20]:

$$(A)_x = \{c \mid c = a + x \forall a \in A\}$$

12.1 Elemento estructurante

Las transformaciones morfológicas están basadas en la utilización de lo que se denomina el elemento estructurante B , el cual está caracterizado por su forma, su tamaño y la localización de su centro. Así, cada pixel dentro de una imagen es comparado con B , haciendo coincidir el centro de B con el pixel. Dependiendo del tipo de transformación morfológica, el valor del pixel es fijado al valor de uno sus vecinos o al promedio de uno o más de estos.

De acuerdo a la técnica desarrollada para realizar la morfología se pueden definir dos tipos de transformaciones morfológicas: regular y recursiva.

En la transformación morfológica regular, se realiza el barrido del elemento estructurante sobre los pixeles de la imagen original para establecer el nuevo valor de cada pixel de la imagen resultante.

En la transformación morfológica recursiva, el barrido del elemento estructurante se hace sobre cada pixel tomando en cuenta los nuevos valores establecidos por el elemento estructurante sobre los pixeles barridos anteriormente. Obviamente las transformaciones morfológicas regulares son más precisas, pero requieren un mayor tiempo de cálculo.

Diversos tipos de elementos estructurantes pueden ser utilizados, dependiendo del tipo de operación morfológica a ser utilizada. Por ejemplo, en un elemento estructurante cuadrado, se incluyen un punto y sus ocho vecinos. Un elemento estructurante en cruz resulta al emplear un punto y sus elementos vecinos arriba, abajo, a su izquierda y a su derecha. Otra posibilidad es emplear un elemento estructurante sobre un punto y sus seis vecinos que conforman un hexágono.

12.2 Erosión

Si A y B son dos conjuntos en el n -espacio E^n (en una imagen bidimensional $n=2$) con elementos a y b :

$$a = (a_1, \dots, a_n) \text{ y } b = (b_1, \dots, b_n)$$

la erosión de A por B denotada por $A \ominus B$ se define como [22]:

$$A \ominus B = \{x \in E^N \mid x + b \in A \forall b \in B\}$$

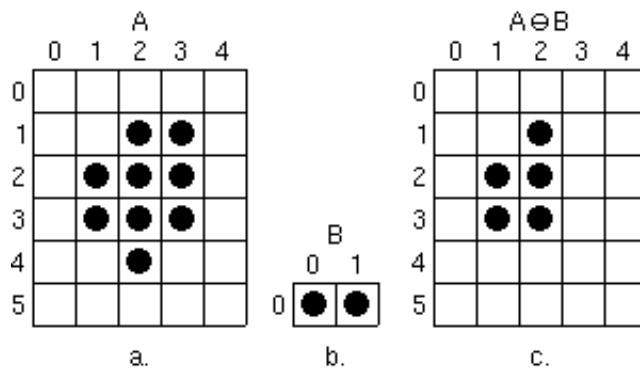
$$A \ominus B = \{x \in E^N \mid x = a - b \forall b \in B \wedge a \in A\}$$

Por ejemplo, la figura 12.1.c presenta la imagen obtenida luego de erosionar la imagen A por el elemento estructurante B mostrados en las figuras 12.1.a y 12.1.b respectivamente.

La erosión de A por el elemento estructurante B puede ser también definida como [38]:

$$A \ominus B = \{c \in E^N \mid (B)_c \subseteq A\}$$

lo cual significa que la erosión de A por B puede ser interpretada como el conjunto de todos los pixeles c tal que el elemento estructurante B , trasladado por c , corresponde a un pixel en A [38].



$$A = \{(1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 2)\}$$

$$B = \{(0, 0), (0, 1)\}$$

$$A \ominus B = \{A + (0, 0)\} \cap \{A + (0, 1)\} = \{(1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 2)\} \cap \{(1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4), (4, 3)\} = \{(1, 3), (2, 2), (2, 3), (3, 2), (3, 3)\}$$

Figura 12.1: Ejemplo de erosión.

En una erosión, los valores del pixel dentro del elemento estructurante son puestos al mínimo valor del elemento. En una imagen en niveles de gris, la erosión por el elemento estructurante B consiste en la búsqueda del valor mínimo de las intensidades dentro de B .

La forma más utilizada para realizar una erosión en una imagen binarizada es la siguiente: se barre la imagen pixel por pixel, al encontrar un pixel blanco, si los pixeles del elemento estructurante son todos blancos el pixel se conservará blanco, de lo contrario devendrá negro. Generalizando, se puede decir, que la erosión consiste en la comparación a nivel local entre un elemento estructurante y un objeto en la imagen que se desea erosionar. Se barre la imagen pixel por pixel desplazando el elemento estructurante sobre ésta. Si al posicionar el elemento estructurante en un punto dado, este se encuentra completamente al interior del objeto el pixel no será transformado; en caso contrario, toma el valor de los pixeles del fondo.

En una imagen binaria la erosión remueve los puntos aislados y las partículas pequeñas, contrae algunas y desconecta otras y elimina picos en el contorno de los objetos. La figura 12.2 presenta el resultado de erosionar por un elemento estructurante de tamaño 3x3 la figura 7.5.



Figura 12.2: Resultado de erosionar la imagen 7.5 por un elemento estructurante de tamaño 3x3.

12.3 Dilatación

La dilatación es una transformación morfológica que combina dos conjuntos usando una adición vectorial entre sus elementos [22]. Si A y B son dos conjuntos en el n -espacio E^n con elementos a y b :

$$a = (a_1, \dots, a_n) \text{ y } b = (b_1, \dots, b_n)$$

la dilatación de A por B denotada por $A \oplus B$ se define como:

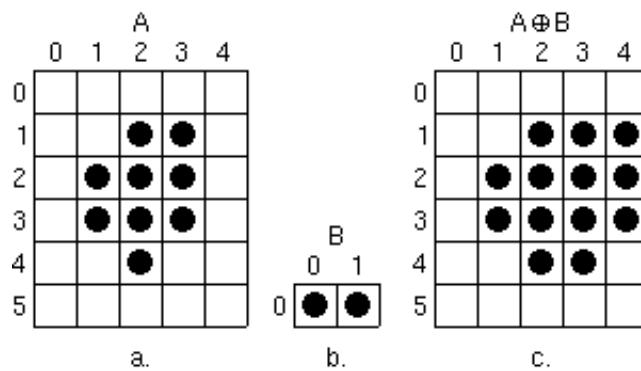
$$A \oplus B = \{c \in E^n \mid x + b \in A \forall b \in B\}$$

responde al conjunto de todas las posibles sumas de los elementos de A con los elementos de B .

La dilatación de A por B puede ser vista como la unión de las translaciones de A especificadas por los elementos del elemento estructurante B [22]:

$$A \oplus B = \bigcup_{b \in B} (A)_b \quad (12.1)$$

La ecuación 12.1 es conocida como la adición de Minkowsky de dos conjuntos [20]. Por ejemplo, al dilatar la imagen A por el elemento estructurante B mostrados en las figuras 12.3.a y 12.3.b respectivamente, se obtiene la imagen resultante presentada en la figura 12.3.c.



$$A = \{(1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 2)\}$$

$$B = \{(0, 0), (0, 1)\}$$

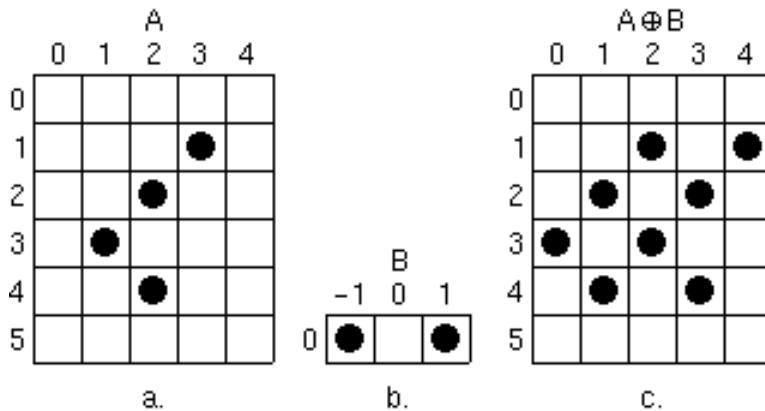
$$A \oplus B = \{(1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4), (4, 2), (4, 3)\}$$

Figura 12.3: Ejemplo 1 de dilatación.

Dado que la dilatación es una operación conmutativa, también puede verse como el resultado de la unión de todas las translaciones del elemento estructurante B definidas por cada pixel de A [38]:

$$A \oplus B = \bigcup_{a \in A} (B)_a$$

La figura 12.4 presenta un ejemplo de un caso particular, en el cual el origen no está incluido en el elemento estructurante B . Como puede observarse en este caso, no necesariamente todos los píxeles de A están incluidos en el resultado. Sin embargo, este caso es no generalmente empleado y no será considerado.



$$\begin{aligned} A &= \{(1,3), (2,2), (3,1), (4,2)\} \\ B &= \{(0,-1), (0,1)\} \\ A \oplus B &= \{(1,2), (1,4), (2,1), (2,3), (3,0), (3,2), (4,1), (4,3)\} \end{aligned}$$

Figura 12.4: Ejemplo 2 de dilatación.

Una dilatación es lo opuesto a una erosión, pero no es su operación inversa. En una dilatación, los valores del pixel dentro del elemento estructurante son colocados al valor máximo de la vecindad del pixel. Así, en una imagen en niveles de gris, la dilatación por el elemento estructurante B consiste en la búsqueda del valor máximo de las intensidades dentro de B .

La forma más utilizada para realizar una dilatación en una imagen binarizada es la siguiente, se barre la imagen pixel por pixel, al encontrar un pixel negro, si al menos uno de los píxeles del elemento estructurante es blanco el pixel devendrá blanco, de lo contrario seguirá siendo negro. Generalizando, se puede decir, que la dilatación consiste en la comparación a nivel local entre un elemento estructurante y un objeto de la imagen a ser dilatado. Se barre la imagen pixel por pixel desplazando el elemento estructurante sobre ésta. Cuando el elemento estructurante es posicionado en un punto dado, en el cual toca el objeto, el pixel es transformado y toma el mismo valor de los píxeles del objeto, en caso contrario permanece con el mismo valor.

Una dilatación aumenta el tamaño de las partículas, llena los pequeños agujeros dentro de ellas y los gollos en el contorno de los objetos y puede conectar partículas vecinas. La figura 12.5 presenta el resultado luego de dilatar la figura 7.5 por un elemento estructurante de tamaño 3x3.

La erosión y la dilatación son reiterativas; esto es, el efecto de repetir una erosión o una dilatación de tamaño 1 n veces es equivalente a desarrollar una erosión simple con un elemento estructurante de tamaño n .

La erosión y la dilatación constituyen los operadores básicos de la morfología matemática, es decir, que a partir de ellos se pueden construir operadores aún más complejos.

12.4 Apertura

El operador morfológico de apertura de un conjunto A por un elemento estructurante B , denotado $A \circ B$ puede expresarse como el resultado de una erosión de A por el elemento estructurante B , seguido por una dilatación por el mismo elemento estructurante B . Expresado matemáticamente:

$$A \circ B = (A \ominus B) \oplus B$$



Figura 12.5: Imagen 7.5 luego de una dilatación por un elemento estructurante 3x3.

Generalizando, se puede decir, que la operación morfológica de apertura en una imagen binarizada, consiste en la comparación a nivel local entre una forma dada por el elemento estructurante y un objeto en la imagen a ser transformado. Se barre la imagen pixel por pixel desplazando el elemento estructurante sobre ésta. Si al posicionar el elemento estructurante en un punto dado, se encuentra completamente al interior del objeto, todos los pixeles no serán transformados y aparecerán en la imagen resultante; en caso contrario, los puntos tomarán el valor de los pixeles del fondo.

Una apertura no destruye tanta información como la erosión. Una apertura desecha los objetos pequeños, pero mantiene los más grandes con una forma muy similar tanto en la imagen original como en la final. Donde la superficie es más pequeña que el elemento estructurante, la apertura remueve pequeños objetos y pequeños detalles en el contorno y puede desconectar algunas partículas y zonas de un objeto. Por lo tanto, la apertura puede ser empleada para separar un objeto de ciertas partes de acuerdo al tamaño del elemento estructurante.

12.5 Cerrado

El operador morfológico de cerrado de un conjunto A por un elemento estructurante B , denotado $A \bullet B$ puede expresarse como el resultado de una dilatación de A seguido por una erosión por el mismo elemento estructurante B . Expresado matemáticamente:

$$A \bullet B = (A \oplus B) \ominus B$$

Una operación de cerrado es similar a una dilatación, pero menos destructivo. Los objetos pequeños pueden ser unidos a los más grandes.

El cerrado llena los huecos o fisuras dentro de las partículas, elimina las pequeñas entradas, pliegues y pequeños detalles del borde de los objetos emparejando el contorno y conecta partículas cercanas. El cerrado es llevado a cabo deslizando el elemento estructurante en el exterior de la partícula y llenando todos los agujeros donde el elemento estructurante no puede ajustar.

El cerrado empareja las partículas más grandes y conecta las pequeñas en una imagen.

12.6 Obtención del contorno de un objeto

Los operadores morfológicos anteriormente vistos pueden ser empleados para obtener el contorno de un objeto A , denotado $\beta(A)$. El contorno obtenido puede ser el externo o el interno dependiendo del método utilizado.

Tal como se muestra en el ejemplo de la figura 12.6, el contorno interior de un objeto binarizado puede ser obtenido fácilmente mediante la siguiente operación morfológica:

$$A - (A \ominus B)$$

es decir, substrayendo de la imagen binarizada original, la imagen erosionada de la misma.

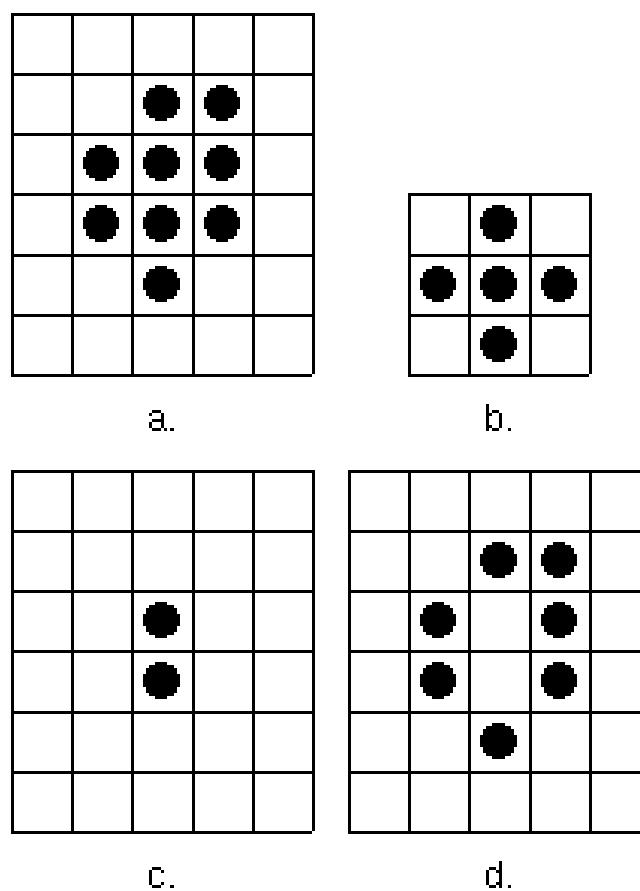


Figura 12.6: Obtención de contorno interno. a. Imagen original. b. Elemento estructurante. c. Imagen erosionada. d. Contorno interno.

De manera similar y tal como se muestra en el ejemplo de la figura 12.7, el contorno externo de un objeto puede obtenerse restando de la imagen dilatada la imagen original:

$$(A \oplus B) - A$$

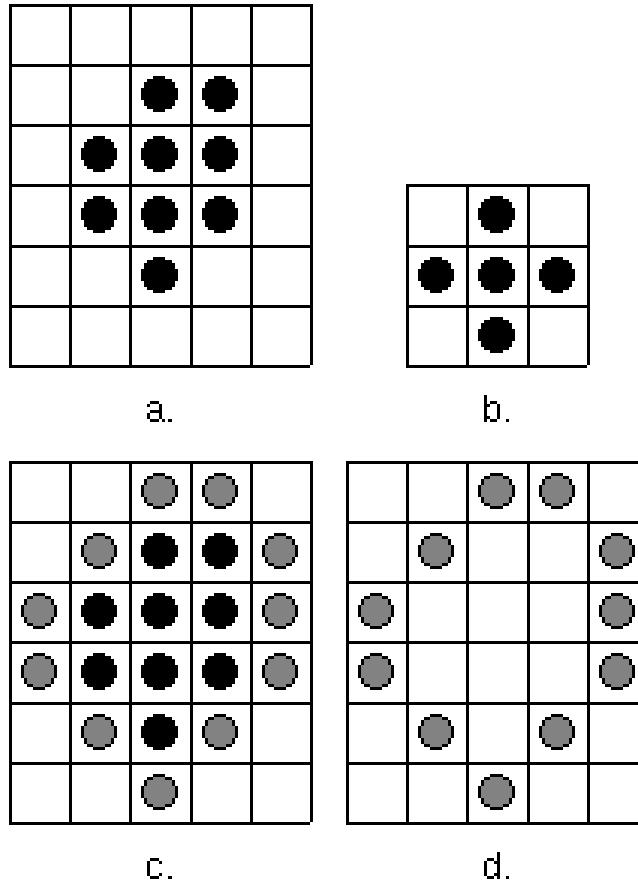


Figura 12.7: Obtención de contorno externo. a. Imagen original. b. Elemento estructurante. c. Imagen dilatada. Contorno externo.

12.6.1 Operador morfológico de gradiente

Este operador morfológico permite también obtener el contorno de un objeto a partir de una estimación de la magnitud del gradiente. El resulta de la composición de tres operaciones. De un lado, se dilata la imagen original. De otro lado, se erosiona la imagen original empleando el mismo elemento estructurante y finalmente se obtiene la substracción de las dos imágenes resultantes:

$$(A \oplus B) - (A \ominus B)$$

12.6.2 Relleno de regiones mediante dilatación condicional

El relleno de regiones consiste en llenar una región completamente delimitada o cerrada que incluya huecos en su interior. Un algoritmo sencillo para la obtención de los pixeles de relleno está basado en técnicas de morfología matemática. Se requiere de la escogencia de un pixel q , situado al interior de la región a llenar A , al cual se asigna un valor de uno y se aplica a continuación el siguiente procedimiento de manera iterativa [20]:

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots \quad (12.2)$$

donde X_k corresponde al conjunto de pixeles de relleno de A , $X_0 = q$ y B es el elemento estructurante mostrado en la figura 12.8.

El algoritmo concluye cuando no hay ningún cambio luego de una iteración, es decir, cuando $X_k = X_{k+1}$.

$X_k \cup A$ permite obtener la región A y su relleno. Esta técnica es denominada dilatación condicional, puesto que al realizar la intersección con A^c en la ecuación 12.2 en cada paso del algoritmo se evita que toda el área de la imagen sea rellenada, haciendo que el relleno se limite al área dentro de la región de interés [20].

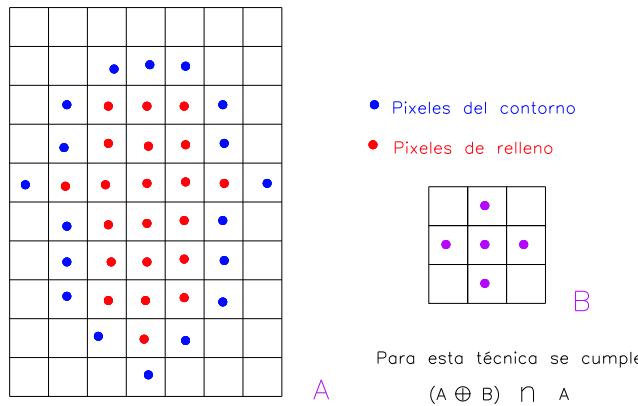


Figura 12.8: Técnica de llenado de regiones.

12.6.3 Detección de elementos conectados

Aplicando el concepto de conectividad es posible la determinación de las regiones definidas por pixeles conectados en una imagen binaria. La extracción de estas regiones es de gran aplicación en el área del procesamiento automático de imágenes. Un algoritmo sencillo para la obtención de pixeles conectados está basado en técnicas de morfología matemática. Se requiere de la escogencia de un pixel q , situado dentro de un componente conectado Y contenido en un conjunto A , el cual tiene un valor de uno y se aplica a continuación el siguiente procedimiento de manera iterativa [20]:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots \quad (12.3)$$

donde X_k corresponde al conjunto de pixeles de relleno de A , $X_0 = q$ y B es un elemento estructurante formado por los 4 u 8-vecinos, mostrados en las figuras 12.8 y 12.9.

El algoritmo concluye cuando no hay ningún cambio luego de una iteración, es decir, cuando $X_k = X_{k+1}$, y se tiene entonces, $Y = X_k$.

La ecuación 12.3 es similar a la ecuación de relleno 12.2. La diferencia radica en el uso de A y no de su complemento, lo cual se debe a que en este caso los pixeles buscados tienen un valor de uno y no de cero [20].

12.7 Filtro sal y pimienta

Con el fin de eliminar algunos pixeles aislados en blanco debidos a ruido, que puedan quedar en el fondo negro de una imagen luego de una operación de binarización, se aplica una operación morfológica conocida como operador sal y pimienta. Este filtro consiste en cambiar un pixel blanco en negro, si se encuentra totalmente rodeado de pixeles negros dentro de una ventana de tamaño 3x3.

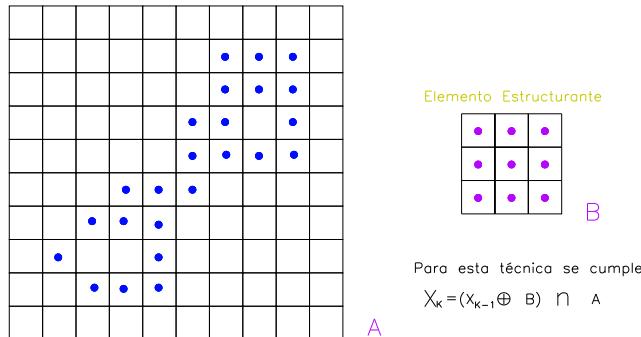


Figura 12.9: Detección de objetos conectados. Adaptado de [20].

12.8 Filtro de mayoría negra

Con el fin de eliminar el ruido que pueda quedar en el fondo, en negro, luego de una operación de binarización de una imagen, se aplica una operación morfológica conocida como operador de mayoría negra [41], el cual es una generalización del filtro sal y pimienta. Este consiste en cambiar un pixel blanco en negro, si la mayoría de los pixeles que lo rodean dentro de una ventana de tamaño 3x3 son negros.

12.9 Adelgazamiento y esqueletización

Una importante aproximación para la representación de una forma estructural o región en una imagen es reducirla a un grafo. Esta reducción puede ser conseguida mediante un procedimiento de adelgazamiento hasta obtener un esqueleto. Esta técnica es utilizada en robótica para darle al robot la indicación de los objetos que debe reconocer y manipular.

Los términos adelgazamiento y esqueletización han llegado a ser casi sinónimos y expresan el procedimiento que transforma una imagen en otra representando su esqueleto [28]. Ver figura 12.10.c. Si el algoritmo escogido es adecuado, la información presente en la imagen original se conserva en su esqueleto. El término esqueleto, también llamado eje medio [3], se refiere en general a la representación de un patrón mediante una colección de curvas o arcos delgados o cercanamente delgados [28].

En general, el esqueleto de una región en una imagen binaria es menos sensible al ruido que cualquier otra técnica de extracción de atributos topológicos [3]. Los algoritmos de esqueletización empleadas sobre las primitivas del contorno entregan representaciones delgadas y robustas. Los esqueletos constituyen una base para la evaluación precisa de los atributos topológicos y métricos de las siluetas. Para obtener una esqueletización adecuada, es indispensable una segmentación también adecuada.

12.9.1 Obtención del esqueleto mediante la transformación del eje medio

El esqueleto de una región puede ser obtenido mediante la transformación del eje medio, técnica denominada *MAT*, por su nombre en inglés *Medial Axis Transformation*, propuesta por Blum [20][38]. La figura 12.10 presenta un ejemplo de esqueletización obtenido por esta técnica.

Sea una región R de una imagen con borde B , para cada pixel q en R se encuentra el pixel vecino más cercano a B . Si q tiene más de uno de dichos vecinos, se dice que pertenece al eje medio o esqueleto de R . El concepto de cercano depende de la definición de distancia empleada y por lo tanto, los resultados de la transformación dependen de ésta.

Para el cálculo del eje medio se siguen entonces los siguientes pasos:

1. Umbralización de la imagen.
2. Aplicar técnica de adelgazamiento.
3. Aplicar la detección de líneas para determinar los pixeles correspondientes al eje medio.

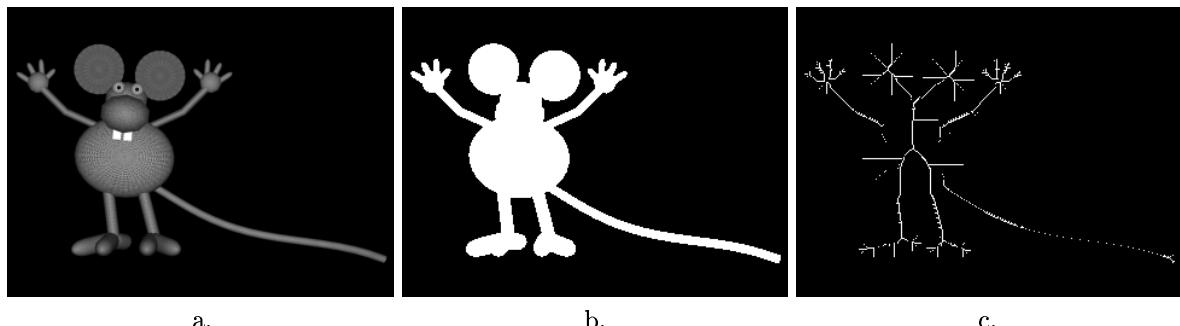


Figura 12.10: Ejemplo de esqueletización obtenida por la transformación del eje medio. a. Imagen original. b. Imagen binarizada. c. Esqueleto.

El esqueleto resultante debe tener las siguientes características:

- Tener un pixel de espesor.
- Atravesar la mitad del objeto.
- Preservar la topología del objeto.

12.9.2 Adelgazamiento

La técnica de adelgazamiento es usada repetidamente para resaltar las líneas centrales con los valores máximos de la distancia al borde del objeto. Un ejemplo de adelgazamiento se presenta en la figura 12.11.

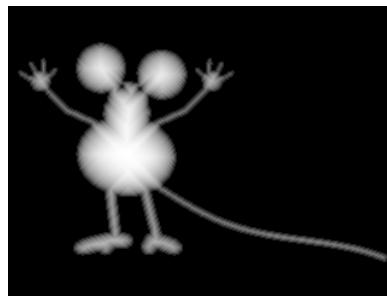


Figura 12.11: Ejemplo de adelgazamiento.

Esta técnica, como se ilustra en la figura 12.12, consiste una imagen de distancias, revisando los 8-vecinos de cada pixel q perteneciente a la región de interés en blanco y se determina si q debe cambiar su valor. El valor final de q representa la distancia más corta entre q y el borde de la región. Así, el valor de q se incrementa si alguna de las condiciones siguientes se cumple:

Los valores de sus pixeles vecinos arriba, abajo, a izquierda o a derecha son iguales a q .

Los valores de sus pixeles vecinos diagonales son iguales a q .

12.9.3 Detección de líneas

La detección de líneas en el proceseo de esqueletización consiste en la selección de los pixeles con los valores máximos de distancia al borde de la región, luego de haberse aplicado la técnica de cálculo de distancias para el adelgazamiento. Al igual que en la técnica de adelgazamiento, en la detección de líneas se analiza cada pixel q en la región de interés y se compara con sus 8-vecinos. Si q cumple algunas de las siguientes condiciones se considera como un punto del eje medio:

- q mayor que sus pixeles vecinos arriba y abajo de él.

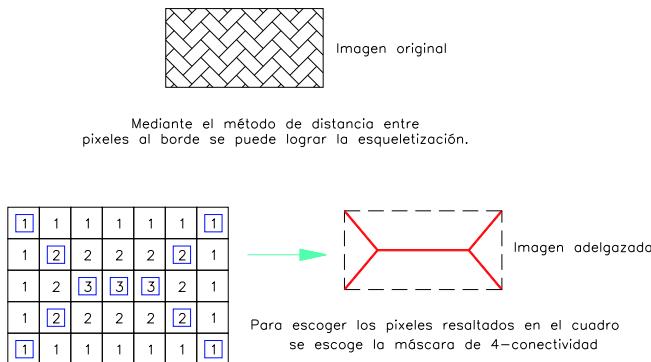


Figura 12.12: Técnica de adelgazamiento.

- q mayor que sus pixeles vecinos a su izquierda y a su derecha.
- q mayor que sus pixeles vecinos en cualquiera de las dos diagonales.

La figura 12.10.c muestra un ejemplo de esta técnica.

12.10 Seguimiento del contorno

El contorno de un conjunto de pixeles conectados C puede ser definido como el conjunto de todos los pixeles de C que tienen al menos un vecino no perteneciente a C .

Las técnicas de seguimiento del contorno permiten obtener los puntos que componen el borde o contorno de un objeto en forma sucesiva. En el algoritmo ilustrado en la figura 12.13, se escoge primero una dirección a seguir en caso de encontrar un pixel perteneciente al objeto cuando se sigue el contorno y se toma la dirección opuesta si se encuentra un pixel fuera del objeto. Suponiendo que si se encuentra el objeto se toma a la izquierda y en caso contrario a la derecha, el algoritmo comienza por barrer la imagen hasta encontrar un pixel del objeto. Luego se sigue la siguiente rutina hasta volver a llegar al pixel de partida: girar a la izquierda e ir hasta el pixel siguiente si se está dentro del objeto; sino girar a la derecha y avanzar nuevamente un pixel.

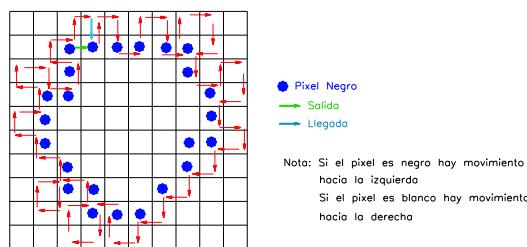


Figura 12.13: Técnica de seguimiento del contorno de un objeto.

Capítulo 13

Transformaciones geométricas de una imagen en el espacio bidimensional

Este capítulo presenta las transformaciones geométricas a que puede ser sometida una imagen en el espacio de dos dimensiones (2D).

13.1 Representación de puntos

En esta sección las imágenes son tratadas como objetos geométricos, ya sea en el plano bidimensional donde cada punto o pixel ubicado por sus coordenadas (x, y) se representa por el vector $[x \ y]$, o como un punto en el espacio tridimensional donde cada punto o pixel se representa por el vector $[x \ y \ z]$.

Para transformar un punto \mathbf{P} , se trabaja con matrices de transformación así por ejemplo, en el caso siguiente un punto ubicado en la coordenada (x, y) es desplazado al punto (x^*, y^*) como se observa a continuación:

$$\mathbf{PT} = [\begin{matrix} x & y \end{matrix}] \left[\begin{matrix} a & b \\ c & d \end{matrix} \right] = [\begin{matrix} (ax + cy) & (bx + dy) \end{matrix}] = [\begin{matrix} x^* & y^* \end{matrix}]$$

donde:

\mathbf{P} es el vector posición del punto origen.

\mathbf{T} es la matriz de transformación.

La convención x^* se refiere al nuevo estado de la coordenada x del punto \mathbf{P} luego del tratamiento realizado. El caso de la transformación identidad está entonces dado por:

$$\mathbf{PT} = [\begin{matrix} x & y \end{matrix}] \left[\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \right] = [\begin{matrix} x & y \end{matrix}] = [\begin{matrix} x^* & y^* \end{matrix}]$$

13.2 Cambio de escala

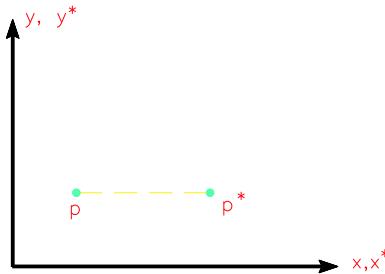
El escalamiento depende de los términos diagonales de la matriz de transformación.

13.2.1 Escalización en x

Para un punto \mathbf{P} definido por sus coordenadas (x, y) , el siguiente manejo matricial produce una escalización del punto en relación a la coordenada x :

$$[\begin{matrix} x & y \end{matrix}] \left[\begin{matrix} a & 0 \\ 0 & 1 \end{matrix} \right] = [\begin{matrix} ax & y \end{matrix}] = [\begin{matrix} x^* & y^* \end{matrix}]$$

En la figura 13.1 se observa la forma en que ocurre esta escalización:

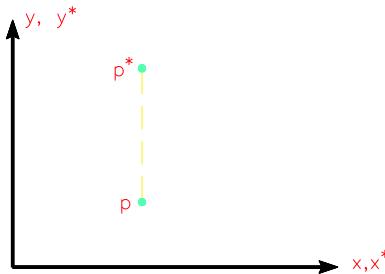
Figura 13.1: Escalización de la abscisa de un punto **P**.

13.2.2 Escalización en y

Para este caso se hace un tratamiento similar al utilizado para la escalización en x :

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & b \end{bmatrix} = \begin{bmatrix} x & by \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

Y en el plano cartesiano, se observa en la figura 13.2.

Figura 13.2: Escalización de la ordenada de un punto **P**.

13.2.3 Forma general de la escalización

Fusionando las dos técnicas de escalización vistas, se tiene:

$$\mathbf{PT} = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} = \begin{bmatrix} ax & dy \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

- Si $a, d > 1$ aumenta el tamaño de la imagen.
- Si $a, d < 1$ disminuye el tamaño de la imagen.
- Si $a \neq d$ se tiene una escalización diferente en x y y de la imagen.

13.3 Reflexión

Al igual que en el escalamiento, la reflexión depende de los términos de las diagonales de la matriz de transformación.

13.3.1 Reflexión alrededor del eje x

Para realizar la reflexión de un pixel alrededor del eje x , se aplica la siguiente transformación sobre el punto:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} x & -y \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

13.3.2 Reflexión alrededor del eje y

La reflexión de un pixel alrededor del eje y , se realiza de manera similar a la transformación anterior aplicando sobre el punto:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -x & y \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

13.3.3 Reflexión alrededor del origen

Como se puede deducir de las dos transformaciones de reflexión anteriores, la reflexión de un pixel alrededor del origen $(0, 0)$, se hace mediante la siguiente transformación:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -x & -y \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

13.4 Deformación

Este tipo de transformación deforma la imagen y es denominada en inglés shear.

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} x & xb + y \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

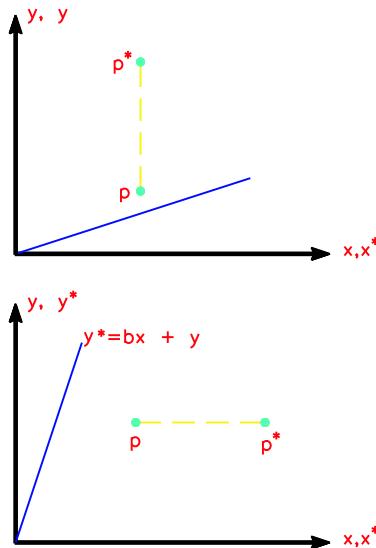


Figura 13.3: Deformación de una imagen.

El caso general para una deformación se da cuando se tiene la siguiente matriz de transformación:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ax + cy & bx + y \end{bmatrix} = \begin{bmatrix} x^* & y^* \end{bmatrix}$$

El único punto que no está sujeto a deformación es el $(0, 0)$, ya que la operación matricial sobre este punto no presenta un resultado diferente al inicial.

13.5 Rotación

La rotación de un pixel un ángulo θ con respecto al eje x se puede realizar como se ve a continuación.

A. Rotación de -90°

$$\mathbf{PT} = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} -y & x \end{bmatrix} \begin{bmatrix} x^* & y^* \end{bmatrix}$$

B. Rotación de -180°

$$\mathbf{PT} = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -x & -y \end{bmatrix} \begin{bmatrix} x^* & y^* \end{bmatrix}$$

C. Rotación de -270°

$$\mathbf{PT} = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} -x & y \end{bmatrix} \begin{bmatrix} x^* & y^* \end{bmatrix}$$

En general, para realizar la rotación de un pixel un ángulo θ cualquiera con respecto al eje x , se aplica la siguiente matriz de transformación:

$$\mathbf{T} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

13.6 Traslación y coordenadas homogéneas

Como se puede observar, el origen del sistema de coordenadas es invariante con respecto a las transformaciones, es decir el punto $(0, 0)$ seguirá siendo el mismo luego de la transformación.

Para trasladar una imagen incluyendo el punto origen, se necesita realizar la siguiente conversión:

$$x^* = ax + cy + m$$

$$y^* = bx + dy + n$$

Como no es posible introducir las constantes de traslación m, n en la matriz de transformación 2x2 utilizada:

$$\mathbf{T} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

se introduce una coordenada adicional llamada coordenada homogénea.

Un vector de posición no homogéneo $[x \ y]$ puede ser representado en coordenadas homogéneas por el vector $[x' \ y' \ h]$ donde $x = \frac{x'}{h}$ y $y = \frac{y'}{h}$, siendo h cualquier número real, donde $h = 0$ tiene un significado especial que no será tratado aquí.

Todas las coordenadas homogéneas son de la forma $[hx \ hy \ h]$ y por lo tanto no hay una única representación de un punto en este tipo de coordenadas, por ejemplo los vectores

$[12 \ 18 \ 3], [8 \ 12 \ 2], [4 \ 6 \ 1]$ representan todos el punto $(4, 6)$.

Sin embargo, por facilidad el valor de h más utilizado es igual a 1 y el grupo de coordenadas de la forma $[x \ y \ 1]$ se escoge para representar el vector $[x \ y]$ en el plano 2-D y la matriz de transformación \mathbf{T} será ahora de 3x3 definida como:

$$\mathbf{T} = \begin{bmatrix} a & b & p \\ c & d & q \\ m & n & 1 \end{bmatrix}$$

Entonces la translación pura se representa en forma matricial por:

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = \begin{bmatrix} x + m & y + n & 1 \end{bmatrix}$$

En el caso donde $p \neq 0$ y $q \neq 0$ puede verse el siguiente efecto:

$$\begin{bmatrix} x^* & y^* & h \end{bmatrix} = \begin{bmatrix} hx & hy & h \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & p \\ 0 & 1 & q \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & px + qy + 1 \end{bmatrix}$$

donde:

- $x^* = x$
- $y^* = y$
- $h = px + qy + 1$

Luego el vector de posición transformado en coordenadas homogéneas $[x^* \ y^* \ h]$ cae ahora en un plano en el espacio 3-D definido por $h = px + qy + 1$. Sin embargo el resultado de interés corresponde al plano físico $h = 1$, pero otros efectos pueden ser obtenidos si se utiliza otro valor de h como se vió antes.

13.7 Rotación alrededor de un punto arbitrario

Las ecuaciones de rotación mostradas anteriormente permiten la rotación de una imagen sólo alrededor del origen. Cuando se desea hacer una rotación alrededor de otro punto, se traslada el origen al punto alrededor del cual se desea rotar la imagen, luego se gira la imagen alrededor del nuevo origen y finalmente se regresa al primer punto origen. Esto se puede expresar matemáticamente en la siguiente forma:

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \operatorname{sen} \theta & 0 \\ -\operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}$$

Las transformaciones matriciales se pueden agrupar con el fin de hacer más eficiente el cálculo cuando el número de puntos a transformar es elevado. Es de anotar que las operaciones entre matrices no son comutables, por ejemplo:

$$\mathbf{X}\mathbf{T}_{Reflexión}\mathbf{T}_{Escalización}\mathbf{T}_{Rotación} \neq \mathbf{X}\mathbf{T}_{Escalización}\mathbf{T}_{Reflexión}\mathbf{T}_{Rotación}$$

Por lo tanto para rotar una imagen alrededor de un punto arbitrario se aplica:

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ -m(\cos \theta - 1) + n \sin \theta & -n(\cos \theta - 1) - m \sin \theta & 1 \end{bmatrix}$$

13.8 Técnicas de mapeo

Para transformar una imagen se pueden emplear dos técnicas diferentes. En la primera, llamada de mapeo hacia adelante, se pintan los pixeles obtenidos de la imagen fuente en la posición apropiada de la imagen destino. Sin embargo, puede suceder que algunos pixeles en la imagen destino permanezcan inmodificados, sin pintarse, debido a que no tienen un correspondiente directo de la imagen fuente y deben ser calculados por interpolación entre los pixeles transformados. Para evitar este problema, se ha desarrollado una segunda técnica, el mapeo inverso, el cual en lugar de examinar qué pixel en la imagen resultante corresponde a un determinado pixel de la original, calcula qué pixel de la imagen original da el valor de intensidad a un determinado pixel de la imagen resultante.

Capítulo 14

Transformaciones geométricas de una imagen en el espacio tridimensional

La modificación de la geometría de una imagen en el espacio tridimensional (3D) se obtiene a partir de la generalización de las ecuaciones obtenidas para las transformaciones 2D. Este tipo de transformaciones puede ser empleada en el desarrollo de técnicas de visualización tridimensional como lo ilustra el ejemplo de la figura 14.1.

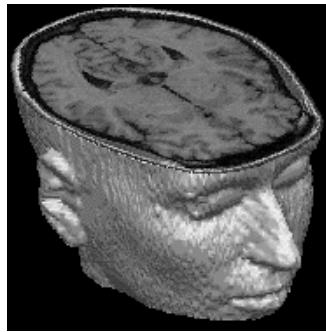


Figura 14.1: Aplicación de la transformación 3D de una imagen.

Luego el punto $[\begin{array}{cccc} x & y & z & h \end{array}]$ puede ser expresado en coordenadas homogéneas como:

$$[\begin{array}{cccc} x' & y' & z' & h \end{array}] = [\begin{array}{cccc} x & y & z & 1 \end{array}]$$

$$[\begin{array}{cccc} x^* & y^* & z^* & 1 \end{array}] = [\begin{array}{cccc} \frac{x'}{h} & \frac{y'}{h} & \frac{z'}{h} & 1 \end{array}]$$

La matriz de transformación será ahora:

$$\mathbf{T} = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{bmatrix}$$

$$\left[\begin{bmatrix} 3 & & & \\ & * & & \\ & & 3 & \\ \hline 1 & * & 3 \end{bmatrix} \quad \begin{bmatrix} 3 \\ * \\ 1 \\ \hline 1 * 1 \end{bmatrix} \right]$$

14.1 Escalización

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} ax & ey & jz & 1 \end{bmatrix} = \begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix}$$

14.2 Deformación

La operación de deformación, denominada shearing en inglés, está definida en la siguiente forma:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ g & i & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x + dy + gz & bx + y + iz & cx + fy + z & 1 \end{bmatrix} = \begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix}$$

14.3 Rotación

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ Alrededor del eje } x$$

$$\mathbf{T} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ Alrededor del eje } y$$

$$\mathbf{T} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ Alrededor del eje } z$$

14.4 Reflexión alrededor de una línea arbitraria

Para rotar una imagen alrededor de una línea arbitraria, se debe seguir el siguiente algoritmo:

1. Trasladar la línea y la imagen de manera que la línea pase por el origen.
2. Rotar la línea y la imagen de manera que la línea coincida con uno de los ejes de coordenadas.
3. Hacer la reflexión alrededor del eje escogido.
4. Aplicar la reflexión inversa.
5. Aplicar la traslación inversa.

Esta transformación puede ser expresada matemáticamente en la siguiente forma:

$$\mathbf{T}\mathbf{R}\mathbf{R}'\mathbf{T}^{-1}$$

donde:

- \mathbf{T} : Matriz de traslación.
- \mathbf{R} : Matriz de rotación.
- \mathbf{R}' : Matriz de reflexión.

14.5 Proyección

Una proyección se puede definir como un tipo particular de transformación que permite convertir una escena o una imagen del espacio 3-D a un plano 2-D que puede ser la superficie de la pantalla de un computador, una hoja de papel, etc como se observa en la figura 14.2.

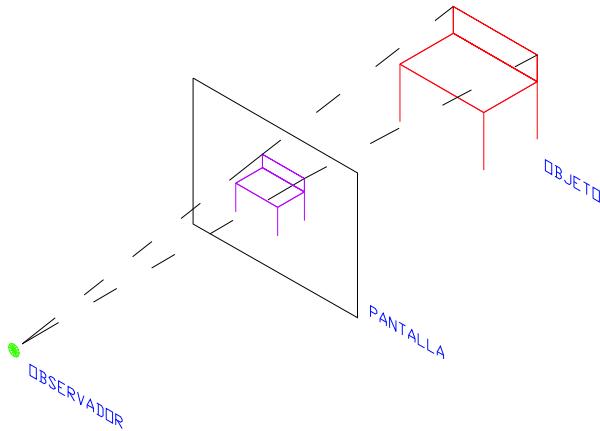


Figura 14.2: Proyección de un objeto sobre la pantalla.

Existen varios tipos de proyecciones, las más utilizadas en computación gráfica son:

- Proyección ortográfica.
- Proyección ortogonal.

14.5.1 Proyección ortográfica

La proyección ortográfica o paralela, es la más simple de las proyecciones y es muy utilizada en ingeniería. De acuerdo con la figura 12.2, se tiene:

$$x_p = x_0$$

$$y_p = y_0$$

Consiste en la proyección del espacio 3-D sobre uno de los planos de coordenadas $x = 0$, $y = 0$ ó $z = 0$. Así las matrices de proyección serán las siguientes:

Para realizar la proyección sobre el plano $x = 0$:

$$\mathbf{P}_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para realizar la proyección sobre el plano $y = 0$:

$$\mathbf{P}_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para realizar la proyección sobre el plano $z = 0$:

$$\mathbf{P}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

14.5.2 Proyección ortogonal

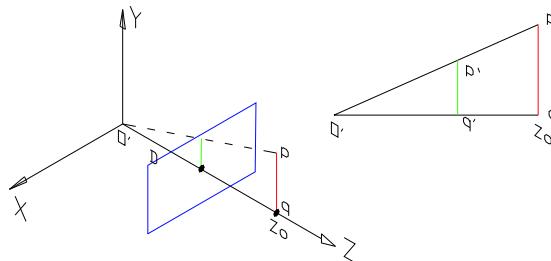


Figura 14.3: Proyección ortogonal.

De acuerdo a la figura 14.3, en la proyección ortogonal se tiene:

$$\frac{pq}{p'q'} = \frac{z_o}{D} = \frac{x_o}{x_p}$$

$$x_p = \frac{x_o D}{z_o}$$

$$y_p = \frac{y_o D}{z_o}$$

Donde (x_p, y_p) son las coordenadas de un punto en la pantalla.

La matriz que produce este resultado está definida por:

$$\mathbf{P}_r = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

14.5.3 Transformación en perspectiva

Cuando uno de los tres primeros elementos de la cuarta columna de la matriz de transformación en coordenadas homogéneas 4x4 no es cero, resulta en una transformación en perspectiva. Esta es una transformación de un espacio 3-D a otro 3-D. A diferencia de lo que ocurre en una proyección en paralelo, el tamaño del objeto se reduce a medida que se incrementa la distancia del objeto al centro de proyección.

Una transformación de un solo punto está dado por:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & rz + 1 \end{bmatrix}$$

En este caso $h = rz + 1 \neq 1$, por consiguiente, las coordenadas ordinarias se obtienen dividiendo las obtenidas por h , luego:

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \begin{bmatrix} \frac{x}{rz+1} & \frac{y}{rz+1} & \frac{z}{rz+1} & 1 \end{bmatrix}$$

Una proyección en perspectiva sobre un plano bidimensional se obtiene aplicando una proyección ortográfica a una transformación en perspectiva.

$$\mathbf{T} = \mathbf{P}_r \mathbf{P}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Luego,

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & 0 & rz + 1 \end{bmatrix}$$

y en coordenadas ordinarias se tiene:

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \begin{bmatrix} \frac{x}{rz+1} & \frac{y}{rz+1} & 0 & 1 \end{bmatrix}$$

Capítulo 15

Composición de imágenes

Las técnicas de composición permiten la superposición de varias imágenes para integrarlas en una sola en la que se pueden apreciar trasparencias y distintas disposiciones de cada imagen con respecto a las otras. Estas son utilizadas en la realización de juegos de video, fusión de imágenes, estereografía, etc.

15.1 Composición sencilla

En muchos juegos de video bidimensionales, los personajes se desplazan sobre un escenario plano empleando lo que en inglés se denomina sprite. Los sprites son pequeñas imágenes correspondientes a los personajes que se mueven sobre una imagen más grande, correspondiente al escenario. En general, los personajes tienen formas irregulares, pero están contenidos en imágenes rectangulares. Por lo tanto, para pintar el personaje sobre la imagen de fondo, una técnica bastante empleada consiste en escoger un color arbitrario que es utilizado como color "transparente". De manera que al pintar el personaje sobre el escenario, solo se reemplaza cada pixel del sprite que no sea transparente sobre la imagen de destino obteniendo el efecto deseado.

15.2 Composite

Otra técnica más compleja pero más potente para la composición de imágenes fue desarrollada por Porter y Duff [16]. Esta técnica denominada en inglés composite permite integrar varias imágenes en una sola mediante la adición de un cuarto valor a la información de cada pixel, conformada por los valores comúnmente utilizados *rojo, verde, azul* (r,g,b) que representan el color de cada punto. Estos tres valores no son suficientes para almacenar el valor de la opacidad o transparencia de un objeto en la imagen, por lo tanto es necesario agregar un canal más de información que permita introducir esta propiedad. Este cuarto valor es llamado canal alpha, de transparencia o mate; por lo cual se tienen los siguientes canales:

$$(r, g, b, \alpha)$$

El valor de alfa representa la opacidad de cada pixel y se encuentra dentro del intervalo $(0, 1)$, donde 0 representa un objeto dentro de la imagen totalmente transparente y 1 un objeto totalmente opaco que al ser colocado sobre otra imagen la cubrirá totalmente (Ver figura 15.1). Un valor intermedio de α indica una opacidad intermedia y por tanto una cobertura parcial de un objeto de una imagen sobre otra.

Hay dos estados particulares en el composite de los cuales es conveniente hacer distinción:

El valor $(0, 0, 0, 0)$ representa un objeto totalmente transparente, mientras que el valor $(0, 0, 0, 1)$ representa un objeto opaco de color negro.

El composite permite reunir dos o más imágenes en una sola, en la que se pueden apreciar transparencias y distintas disposiciones de una imagen con respecto a otra.

Debido a que la imagen final no se trata como una unidad, se hace necesario tomar en cuenta algunas características que no se necesitaban antes. Una de ellas es la transparencia: La cantidad de fondo que se puede

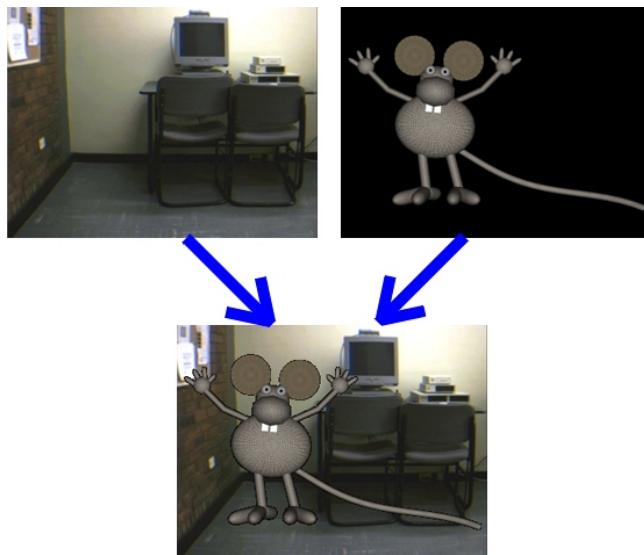


Figura 15.1: Composite entre 2 imágenes.

ver a través de una imagen y la otra es la disposición de las imágenes, es decir, el orden en que se colocan los distintos elementos para producir la imagen final.

15.3 Canal alfa

Cuando se requiere realizar el composite de diversas imágenes es necesario implementar el valor de alfa para cada pixel. Esto se logra multiplicando el valor de cada elemento dentro de la tripleta (r,g,b) por el valor de alfa.

$$\alpha.(r, g, b) = (\alpha.r, \alpha.g, \alpha.b)$$

Cuando se aplica el valor de α a una imagen, se debe almacenar el valor del color multiplicado por α en lugar del color verdadero. Debido a que α varía entre 0 y 1, al hacer la multiplicación, el brillo de la imagen sufre un desplazamiento hacia el negro.

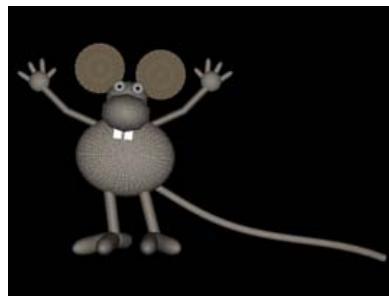


Figura 15.2: Imagen original, $\alpha = 1$.

La premultiplicación del valor α y el desplazamiento del brillo producen la impresión de que la imagen se torna cada vez más opaca. Cuando se aplica un valor de transparencia $\alpha = 0$ sobre un pixel, la multiplicación hace que todos los elementos del color (r, g, b) sean iguales a 0 y desaparezca el pixel de la imagen resultante debido a que el pixel es ahora totalmente transparente.

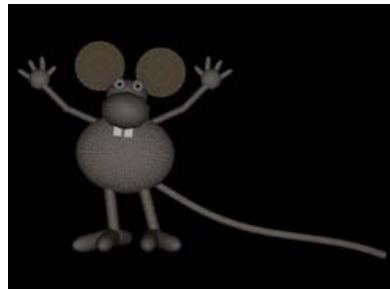


Figura 15.3: Corrimiento del brillo de la imagen anterior hacia el negro por efecto del factor α , $\alpha = 0.7$.

15.4 Algebra del composite

Suponiendo las tres imágenes de la figura 13.5: A (imagen roja), B (imagen azul) y 0 (imagen de fondo), que presentan las siguientes características:

- α_A representa la opacidad de A , por lo tanto, $(1 - \alpha_A)$ representa la transparencia de A , es decir, la cantidad de fondo que se puede observar a través de ella.
- α_B representa la opacidad de B , por lo tanto, $(1 - \alpha_B)$ representa la transparencia de B , es decir, la cantidad de fondo que se puede observar a través de ella.
- Si las imágenes están colocadas una sobre otra, se puede observar que el fondo sólo puede observarse a través de la zona cubierta por $(1 - \alpha_A)(1 - \alpha_B)$.
- $\alpha_A(1 - \alpha_B)$ representa la cantidad de imagen que es bloqueada por A y se puede ver a través de B .
- $\alpha_B(1 - \alpha_A)$ representa la cantidad de imagen que es bloqueada por B y se puede ver a través de A .
- $\alpha_A \cdot \alpha_B$ representa la cantidad de imagen que es bloqueada simultáneamente por A y B .



Figura 15.4: Superposición de imágenes en el composite.

De donde se puede deducir la tabla siguiente:

Descripción	Escogencia
$A \cap \bar{B}$	0
$A \cap B$	$0, A$
$\bar{A} \cap B$	$0, B$
$A \cap B$	$0, A, B$

Uno de los términos más importantes dentro de el composite está representado por las fracciones de la imagen, es decir, la parte de la imagen que subsiste en la imagen final. Por ejemplo, para colocar la imagen A sobre la imagen B , $F_A = 1$ y $F_B = 1 - \alpha_B$. Para cada una de las operaciones del composite que se muestran a continuación, se desea computar la contribución de cada imagen en cada pixel. Para cada figura se asigna una fracción que prevalece hasta la imagen final.

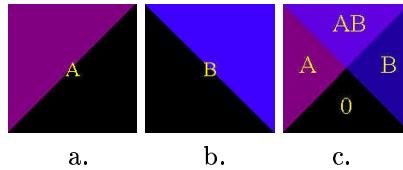


Figura 15.5: Imagenes semitransparentes. Se supone transparente la zona en color negro. a.Imagen A. b. Imagen B.

c. Superposición de las imágenes A y B.

Suponiendo las imágenes semitransparentes A y B mostradas en las figuras 15.5.a y 15.5.b.

Si se coloca una sobre la otra, sin importar el orden, se obtiene la imagen mostrada en la figura 15.5.c, la cual se divide en cuatro regiones:

- 0: Región correspondiente al fondo, en la que no se encuentra ni A ni B .
- A : Región en donde únicamente se encuentra A (no se tiene en cuenta la intersección con B).
- B : Región en donde únicamente se encuentra B (no se tiene en cuenta la intersección con A).
- AB : Región en la que se intersectan las dos imágenes A y B .

Así, se obtiene la siguiente tabla de áreas para las imágenes A y B :

A	B	Nombre	Descripción	Transparencia
0	0	0	$A \cap B$	0
1	0	A	$A \cap B$	0, A
0	1	B	$A \cap B$	0, B
1	1	$A.B$	$A \cap B$	0, A, B

Una operación composite binaria particular puede entonces, ser identificada por una cuádrupla indicando la imagen entrante que contribuye al composite en cada una de las cuatro áreas de subpixeles (0, A , B , AB) de la tabla anterior. Con tres posibilidades de escogencia donde las imágenes se intersectan, dos donde sólo una imagen existe y una donde no se incluye ninguna imagen, se obtienen 12 diferentes operaciones de composite.

La figura 15.6 resume todas las posibles combinaciones de dos imágenes sobre la pantalla.

	O	A	B	AB	DIAGRAMA	FA	FB
Clear	0	0	0	0		0	0
A	0	A	0	A		1	0
B	0	0	B	B		0	1
A over B	0	A	B	A		1	$1 - \alpha_A$
B over A	0	A	B	B		$1 - \alpha_B$	1
A in B	0	0	0	A		α_A	0
B in A	0	0	0	B		0	α_B
A out B	0	A	0	0		$1 - \alpha_B$	0
B out A	0	0	B	0		0	$1 - \alpha_A$
A atop B	0	0	B	A		α_B	$1 - \alpha_A$
B atop A	0	A	0	B		$1 - \alpha_B$	α_A
A xor B	0	A	B	0		$1 - \alpha_B$	$1 - \alpha_A$

Figura 15.6: Operaciones de composite (Adaptada de [16]).

Para cada uno de los operadores se presenta la siguiente información:

- Nombre del Operador.

- Cuádrupla $(0, A, B, AB)$ que representa la imagen que debe colocarse en cada una de las regiones mencionadas anteriormente.
- Representación gráfica.
- Fracción de $A F_A$: Representa la fracción de la imagen A que subsiste en la imagen final.
- Fracción de $B F_B$: Representa la fracción de la imagen B que sobrevive en la imagen final.

15.5 Realización del composite

Para unir dos imágenes A y B por medio del composite se emplean dos fórmulas. La primera ecuación permite encontrar el valor de alfa de la imagen resultante α_c :

$$\alpha_C = \alpha_A F_A + \alpha_B F_B$$

La segunda ecuación ?? permite hallar el valor de cada componente de color de la imagen final c_C :

$$c_C = c_A F_A + c_B F_B$$

C_A, C_B, C_C son los componentes de color originales de cada imagen y c_A, c_B, c_C son los componentes de color luego de premultiplicarlos por α , por ejemplo:

$$c_A = \alpha_A \cdot C_A$$

15.6 Operadores del composite

Existen algunas operaciones que permiten afectar la imagen original y aplicar sobre ella efectos significativos para unirla con otras.

15.6.1 Factor de desvanecimiento

El factor de desvanecimiento δ (en inglés dissolve) permite desvanecer gradualmente una imagen con respecto a otra. Este operador se define:

$$Dissolve(A, \delta) = (\delta.r_A, \delta.g_A, \delta.b_A, \delta.\alpha_A)$$

donde normalmente:

$$0 \leq \delta \leq 1$$

El operador de desvanecimiento desvanece la imagen a medida que δ pasa de 1 a 0.

15.6.2 Factor de oscurecimiento

El factor de oscurecimiento ϕ (en inglés darken) permite oscurecer una imagen sin afectar su transparencia. Este operador está definido en la siguiente forma:

$$Darken(A, \phi) = (\phi.r_A, \phi.g_A, \phi.b_A, \alpha_A)$$

donde normalmente:

$$0 \leq \phi \leq 1$$

15.6.3 Factor de opacidad

El operador de opacidad ω (en inglés opaque) permite disolver una imagen sin afectar su color verdadero:

$$\text{Opaque}(A, \omega) = (r_A, g_A, b_A, \omega \cdot \alpha_A)$$

donde:

$$0 \leq \phi \leq 1$$

15.6.4 Operador de adición

El operador de adición, resumido en la figura 15.7, suma dos imágenes, es decir no monta una sobre la otra.

	O	A	B	AB	DIAGRAMA	FA	FB
A plus B	O	A	B	AB		1	1

Figura 15.7: Operador de adición

Mediante el uso de este operador y el operador de desvanecimiento es posible disolver una imagen en otra:

$$\text{Dissolve}(A, \alpha) + \text{Dissolve}(B, 1 - \alpha)$$

es decir:

$$q = \alpha \cdot q_A + (1 - \alpha) \cdot q_B$$

Capítulo 16

Transformada de Karhunen-Loève

Una de las transformadas empleadas en el procesamiento de imágenes es la transformada de Karhunen-Loève (KLT), también llamada Análisis de Componentes Principales, Transformada Hotelling y aproximación por vectores propios. Esta transformada es calculada a partir de un conjunto de vectores [20][24], basada en las propiedades estadísticas de éstos. Su aplicación principal se da en la compresión de imágenes y la traslación de datos.

Sea una población de vectores aleatorios \mathbf{X} de dimensión d de la forma:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

El vector medio de esta población se define como:

$$\mathbf{m}_x = E(\mathbf{X})$$

donde $E(\mathbf{X})$ es el valor esperado de la población de vectores \mathbf{X} . El cálculo del valor esperado de un vector se realiza evaluando los valores esperados de cada componente del vector.

La matriz de covarianza de la población de vectores \mathbf{X} es definida como:

$$\mathbf{C}_x = E \left\{ (\mathbf{X} - \mathbf{m}_x) (\mathbf{X} - \mathbf{m}_x)^T \right\}$$

Dado que \mathbf{X} es de dimensión d , la matriz de covarianza \mathbf{C}_x es de orden $d \times d$. El elemento c_{ii} corresponde a la varianza del i -ésimo elemento x_i de la población de vectores \mathbf{X} y el elemento c_{ij} a la covarianza entre los elementos x_i y x_j de la población.

\mathbf{C}_x tiene varias propiedades útiles, es real y simétrica. Además, si sus elementos x_i y x_j no están correlacionados, su covarianza es cero, es decir, $c_{ij} = c_{ji} = 0$.

Cuando se tienen n vectores de muestra de una población aleatoria, el vector medio puede ser aproximado por:

$$\mathbf{m}_x = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$$

y la matriz de covarianza por:

$$\mathbf{C}_x = \frac{1}{n} \sum_{k=1}^n \mathbf{X}_k \mathbf{X}_k^T - \mathbf{m}_x \mathbf{m}_x^T$$

Dado que \mathbf{C}_x es real y simétrica, es siempre posible encontrar un conjunto de d vectores propios ortonormales, los cuales permiten conformar una nueva base donde la matriz con respecto a ésta sea tan simple como sea posible. Los vectores forman así un nuevo sistema de coordenadas cuyo origen está en el centro de la población de vectores y donde los ejes están orientados en la dirección de los vectores propios de \mathbf{C}_x .

Se dice que una matriz \mathbf{C} de tamaño $d \times d$ tiene un vector propio \mathbf{Q} y su valor propio correspondiente λ si:

$$\mathbf{C} \cdot \mathbf{Q} = \lambda \mathbf{Q}$$

$$\mathbf{Q} \neq 0$$

Se define λ como valor propio de:

$$\mathbf{C}_x \Leftrightarrow \det(\mathbf{C}_x - \lambda \mathbf{I}) = 0 \quad (16.1)$$

donde \mathbf{I} es la matriz identidad.

A partir de la ecuación 16.1 se obtienen los vectores propios con sus correspondientes valores propios. Se definen e_i como los vectores propios de \mathbf{C}_x con valores propios λ_i , con $i = 1, \dots, d$. Por conveniencia, los vectores propios son colocados en orden descendente de acuerdo a los valores propios, es decir, $\lambda_k \geq \lambda_{k+1}$, para $k = 1, \dots, n-1$.

\mathbf{A} es una matriz de talla $d \times d$ cuyas filas están formadas con los vectores propios de \mathbf{C}_x . De esta manera la primera fila contiene el vector propio con el valor propio más grande y la última fila contiene el vector propio correspondiente al menor de los valores propios. A continuación se construye un vector \mathbf{Y} de la siguiente forma:

$$\mathbf{Y} = \mathbf{A}(\mathbf{X} - \mathbf{m}_x) \quad (16.2)$$

La ecuación 16.2 es llamada Transformada Hotelling.

El vector resultante \mathbf{Y} tiene como propiedad $\mathbf{M}_y = 0$. La matriz de covarianza \mathbf{C}_y del vector \mathbf{Y} se obtiene en términos de \mathbf{A} y \mathbf{C}_x haciendo:

$$\mathbf{C}_y = \mathbf{A} \mathbf{C}_x \mathbf{A}^T$$

Es interesante observar que \mathbf{C}_y es una matriz diagonal, cuyos elementos a lo largo de la diagonal principal son los valores propios de \mathbf{C}_x . Así \mathbf{C}_y está constituido de la siguiente forma:

$$\mathbf{C}_y = \begin{pmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & \lambda_n \end{pmatrix}$$

Una propiedad importante de la transformada de Hotelling se refiere a la reconstrucción de \mathbf{X} a partir de \mathbf{Y} . Debido a que las filas de \mathbf{A} son vectores ortonormales, $\mathbf{A}^{-1} = \mathbf{A}^T$, por consiguiente:

$$\mathbf{X} = \mathbf{A}^T \mathbf{Y} + \mathbf{m}_x$$

En lugar de usar todos los vectores propios de \mathbf{C}_x , se emplean únicamente los k vectores propios con los valores propios más altos dando como resultado una matriz \mathbf{A}_k y los vectores \mathbf{Y} serán k -dimensionales.

El vector reconstruido empleando \mathbf{A}_k es:

$$\widehat{\mathbf{X}} = \mathbf{A}_k^T \mathbf{Y} + \mathbf{m}_x$$

El error introducido al no utilizar los vectores propios correspondientes a los valores propios menos significativos puede ser calculado con precisión usando los valores propios de \mathbf{C}_x . Así, el error cuadrático medio e_{ms} está dado por:

$$e_{ms} = \sum_{j=1}^n \lambda_j - \sum_{j=1}^k \lambda_j = \sum_{j=k+1}^n \lambda_j$$

Este error es minimizado al seleccionar los k vectores propios correspondientes a los valores propios más grandes. Por consiguiente, la transformada Hotelling es óptima con respecto a la minimización del error cuadrático medio entre el conjunto de vectores \mathbf{X} y su aproximación $\widehat{\mathbf{X}}$.

Capítulo 17

Fusión de imágenes mediante el uso de sinogramas

17.1 Introducción

La fusión de imágenes es ampliamente requerida como parte del análisis de pruebas médicas (estudio de plasticidad, localización de tumores u obstrucciones, análisis de resonancia magnética funcional, etc.), construcción de imágenes espaciales o ajuste de imágenes satelitales. Dependiendo del caso particular, se puede requerir fusionar imágenes del mismo tipo que difieren en cuanto a las características del objeto de interés o imágenes obtenidas a través de técnicas diferentes, en las cuales no solo se nota un cambio de posición del objeto, sino que también varía la forma en que presentan la información.

Múltiples técnicas de fusión de imágenes médicas han sido propuestas por diferentes autores. El problema se aborda en algunos de estos casos desde el momento mismo de la adquisición, mediante el uso de "fantasmas" externos de alta intensidad respecto a los cuales se analiza la posición del objeto [39] y se realizan las correcciones necesarias. En otros casos se utilizan características externas del organo a estudiar, la cabeza por ejemplo, o la localización de órganos internos fácilmente identificables que sirven como referencia [29]; en otros resulta muy apropiado el uso de figuras geométricas que se aproximen a la forma del objeto de interés, por ejemplo una ellipse ajustada al corazón [43]. En este último caso las primeras aproximaciones se hacen generalmente a mano.

Métodos de búsqueda como los algoritmos genéticos [25] han sido implementados, junto con muchos otros, con el mismo fin de lograr una buena fusión.

El objetivo final de la mayor parte de procedimientos desarrollados es lograr exactitud, robustez, disminuir los requerimientos computacionales y una amplia posibilidad de uso de los algoritmos.

Este capítulo presenta una técnica basada en la utilización de la transformada de Radón, la cual es una generalización de la transformada de Hough.

17.2 Métodos de proyección

El problema de la manipulación de imágenes comienza con el mismo proceso de adquisición. Se espera que cada resultado proporcione la mayor información con la más alta fidelidad. Procedimientos como las imágenes adquiridas por resonancia magnética (IRM) o el escáner X se utilizan para obtener algunos cortes del objeto de interés. Para construir una imagen, se toma información de cada plano examinando las intensidades de la señal a lo largo de varios rayos paralelos lanzados sobre el objeto. El resultado de cada uno de estos barridos recibe el nombre de proyección.

Este mismo principio puede ser empleado para realizar la fusión de imágenes, partiendo de imágenes previamente adquiridas sobre las cuales se realizan barridos, ya no con rayos físicos, sino haciendo un análisis de los pixeles.

17.2.1 Proyección de Máxima Intensidad

El algoritmo de proyección de máxima intensidad (MIP por sus iniciales en inglés) es usado en general para construir angiogramas bi o tridimensionales identificando altas intensidades dentro de una pila de cortes de IRM [47].

El MIP se construye haciendo un barrido de la(s) imagen(es) mediante líneas paralelas igualmente espaciadas (ver figura 17.1); cada uno de los pixeles que se recorren se examina para tomar el que corresponda a la máxima intensidad. Con los datos obtenidos con cada línea se construye una proyección. En el caso de reconstrucciones tridimensionales, las proyecciones se calculan a diferentes ángulos con el fin de que los datos de profundidad no se pierdan [47].

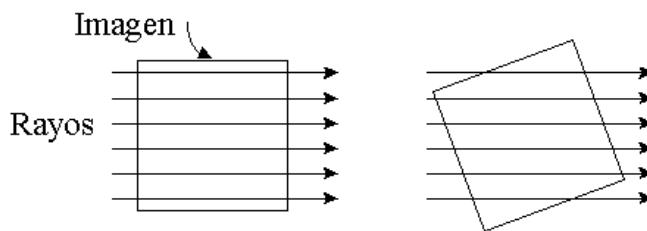


Figura 17.1: Construcción de la Proyección de Máxima Intensidad (MIP).

Del algoritmo estudiado solo se toma el concepto básico de obtención del pixel de máxima intensidad aplicado sobre imágenes bidimensionales, lo que da como resultado proyecciones unidimensionales que son utilizadas posteriormente como elementos comparativos para realizar el ajuste.

Dado que el objetivo es hacer una representación de la imagen que la describa plenamente y que proporcione información fácil de analizar, de forma que permita compararla y relacionarla con la otra que se diferencia de la primera básicamente por estar rotada, los resultados de proyecciones a diferentes ángulos (los necesarios) resulta ser una buena alternativa.

17.2.2 Promedio

La técnica de promedio tiene un principio similar al MIP y al igual que este el objetivo es lograr una identificación de la imagen, la diferencia radica en que en lugar de buscar a lo largo de cada línea de barrido el pixel de máxima intensidad, la proyección se obtiene al promediar las intensidades de los pixeles examinados por cada línea. Los resultados del promedio se aproximan al entero más próximo con el fin de mantener una información coherente.

En la figura 17.2 se muestra el procedimiento seguido en la retroproyección sobre una imagen oscura. En este caso, para obtener la proyección los rayos de inclinan un ángulo θ con respecto a los ejes imaginarios x y y lo que permite tener una vista diferente del objeto.

17.3 Sinograma o transformada de Radón

Una vez obtenidas las proyecciones, se hace necesario organizar los resultados de tal forma que se puedan manipular en forma conveniente en el proceso de análisis.

Cada una de las vistas o proyecciones es una función de la posición del objeto dentro de la imagen y del ángulo de inclinación de los rayos de inspección. Puede entonces construirse una colección bidimensional ordenada de proyecciones en la cual uno de los ejes corresponde a la posición y el otro al ángulo. Esta estructura recibe el nombre de sinograma. Su nombre se debe a la variación sinusoidal de la posición del objeto dentro de las proyecciones debido a la variación consecutiva de los ángulos de adquisición (Ver figura 17.3) [44]. El proyectar los pixeles de máxima intensidad o el promedio del valor de los pixeles correspondientes para diferentes grados, hace que se delineen un movimiento sinusoidal sobre el sinograma, este fenómeno es el que da el nombre de sinograma al conjunto de proyecciones.

Los principios de este método fueron publicados por primera vez en 1917 por Radón (de donde toma el nombre de transformada de Radón), los cuales fueron utilizados luego como herramienta para la reconstrucción de imágenes.

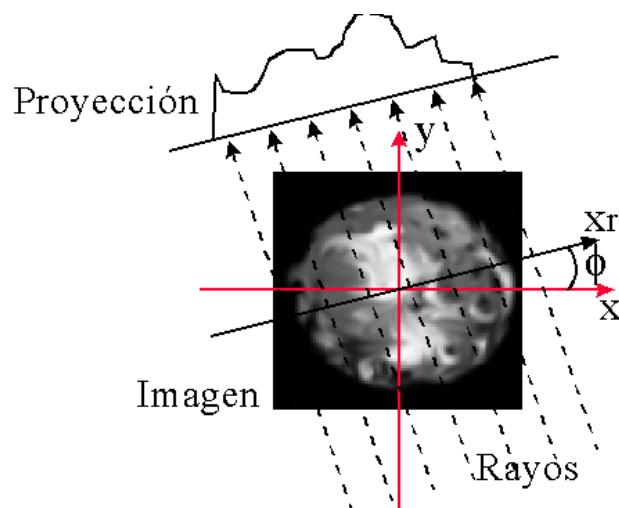


Figura 17.2: Resultado de una proyección con rayos inclinados θ grados.



Figura 17.3: Sinograma obtenido por la técnica MIP de la figura 17.2 haciendo 180 variaciones de 1 grado.

17.4 Correlación

El análisis de correlación es una técnica usada para medir el grado de relación que existe entre variables aleatorias independientes, es decir, el resultado del coeficiente de correlación indica el valor de ajuste entre las variables, aunque esta relación no tenga necesariamente un significado real; no indica si el cambio de una variable causa el cambio en otra, sino si el patrón de cambio de una está relacionado con el patrón de cambio de la otra.

Se basa en los resultados del método de regresión lineal, de forma que "representa la dispersión media de los puntos representativos en torno a la recta de regresión" [31]. El coeficiente de correlación r ha sido definido a través de diferentes expresiones. El coeficiente de correlación normalizado está dado por [31][14]:

$$r = \frac{\sum f(x)f(y)}{\sqrt{\sum f(x)^2 \sum f(y)^2}}$$

y el coeficiente de correlación de Karl Person, por [4]:

$$r = \frac{n \sum xy - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

En este último caso es necesario prestar especial atención en el manejo del número de valores correlacionados n para obtener resultados satisfactorios.

El coeficiente de correlación toma valores entre -1 y 1. El signo del resultado indica si la inclinación de la recta de relación es positiva o negativa y la magnitud indica el grado de asociación que hay entre las variables comparadas.

Para el caso de imágenes, las variables a relacionar son las intensidades resultantes de los métodos de retro-proyección almacenadas en los sinogramas correspondientes, por lo tanto el valor del coeficiente de correlación siempre es positivo de forma que la información que se manipula se restringe al nivel de relación entre las proyecciones comparadas.

17.5 Procedimientos

Tres etapas conforman el proceso de fusión: la obtención del mapa de bits de cada una de las imágenes, la construcción de los sinogramas correspondientes y por último la aplicación del método de correlación deseado.

Para realizar la fusión se necesitan dos imágenes, una imagen de referencia y una imagen flotante; esta última es la imagen a ajustar.

El criterio de transformación geométrica exige definir [10]:

Una imagen de referencia y una imagen flotante.

Una n -upla de parámetros de la transformación (en este caso solo el ángulo de rotación en el plano x, y).

La transformación geométrica que relaciona los pixeles.

Las funciones que relacionan el nivel de gris de los pixeles de las imágenes flotante y de referencia. Para construir el sinograma se utilizan 180 proyecciones obtenidas aplicando alguno de los dos métodos de retro-proyección anteriormente expuestos. Se utiliza un rayo por cada fila de pixeles recorriendo la imagen en sentido horizontal.

Con el fin de obtener el sinograma, se considera un primer procedimiento que consiste en variar el ángulo de inclinación de los rayos entre 0° y 180° . Para implementarlo es necesario utilizar un algoritmo de construcción de líneas que permita identificar los pixeles que cada uno de los rayos debe examinar. Para tal efecto es posible considerar el algoritmo de línea de punto medio de Bresenham [11] y sus variaciones requeridas para construir los caminos en diferentes direcciones.

Considerando que este procedimiento es necesario repetirlo para todos los ángulos que requiere la construcción del sinograma, es posible utilizar los resultados de Didon [15], también afirmados por Russ [44], según los cuales las proyecciones obtenidas al utilizar líneas de búsqueda con una inclinación diferente cada vez, son equivalentes a aquellos que se logran al hacer proyecciones siempre con la misma inclinación de los rayos (0°)

sobre imágenes rotadas. Esta opción reduce el número de cálculos, lo cual favorece la disminución del tiempo de procesamiento.

Para realizar la rotación de la imagen se utilizan las matrices de transformaciones geométricas convencionales.

En primer lugar es necesario trasladar los ejes coordenados al centro de la imagen para hacer la rotación sobre el punto medio de la misma y luego regresar los ejes a su posición inicial. La secuencia de matrices utilizadas para esto es:

MATRÍZ DE TRANSLACIÓN

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}$$

donde m, n son las coordenadas del centro de la imagen.

MATRÍZ DE TRANSLACIÓN INVERSA

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix}$$

MATRÍZ DE ROTACIÓN

$$\begin{bmatrix} \cos \theta & \operatorname{sen} \theta & 0 \\ -\operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Siendo $[x, y, z]$ el punto de partida de la matriz original y $[x*, y*, z*]$ el punto correspondiente a la matriz rotada, se tiene :

$$[x*, y*, z*] = [x, y, z] * \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ m * \cos \theta - n * \sin \theta - m & m * \sin \theta - n * \cos \theta - n & 1 \end{bmatrix}$$

Para garantizar que todos los pixeles en la imagen rotada tengan un valor de intensidad asignado, es necesario hacer un mapeo inverso. Los pixeles que no encuentran uno correspondiente toman el valor de intensidad 0 (negro). El sistema matricial que se ajusta a este caso es:

$$[x, y, z] = [x*, y*, z*] * \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ m - n * \sin \theta - m \cos \theta & n + m * \sin \theta - n * \cos \theta & 1 \end{bmatrix}$$

La mayor parte de los valores resultantes de aplicar esta transformación no son enteros, lo que hace necesario utilizar un método de aproximación. Puede pensarse en usar el valor entero más próximo al resultado. Si se quisiera ser más estricto en la alta calidad de la imagen resultante se puede pensar en interpolar también las intensidades.

Para la construcción del sinograma la imagen de referencia se rota entre 0° y 180° aumentando un grado a la vez. En cada caso se obtiene una proyección usando un rayo horizontal por cada fila del mapa de bits; el resultado se almacena en una matriz de 180 filas (una para cada ángulo) y el número de columnas igual al número de rayos.

Es necesario anotar que el cálculo de proyecciones, y por ende el sinograma, para los demás ángulos es redundante.

Para la imagen flotante sólo se toma la proyección a 0° , con la cual se realizan las operaciones necesarias para correlacionar las dos imágenes.

Una vez construido el sinograma para la imagen de referencia, por medio del coeficiente de correlación, se comparan los resultados almacenados en él con la proyección que se obtuvo de la imagen flotante. Las variables aleatorias a considerar son, por supuesto, las intensidades almacenadas para cada proyección.

Bajo estas consideraciones la correlación se realiza con respecto a los valores que corresponden a estos ángulos, aquel en que el valor de correlación sea máximo es asumido como ángulo de rotación para fusionar las dos imágenes.

Capítulo 18

Metamorfosis

La metamorfosis de imágenes, conocida por el término inglés morphing, es una técnica de transformación morfológica de una imagen a otra. Esta consiste en la creación de imágenes intermedias en las cuales una imagen se transforma gradualmente en una segunda.

El morphing puede ser dividido en dos categorías:

- Metamorfosis de transición que comprende la transformación de una imagen en otra completamente diferente.
- Metamorfosis de deformación o distorsión donde sólo se utiliza una imagen o escena, estirándola o comprimiéndola para dar la ilusión de movimiento. Este tipo de metamorfosis también es conocido como Warping [35].

Estos efectos son utilizados principalmente en la industria del entretenimiento tal como comerciales de televisión, vídeos musicales y películas [11] y también han sido empleados en aplicaciones médicas, tales como la fusión de imágenes.

18.1 Historia

En los años 60, el warping de imágenes fue empleado para manipular imágenes científicas. Esta técnica se utilizó para transformar las imágenes satelitales con el fin de compensar los errores producidos por los sensores o debidos a la curvatura de la tierra [35].

En 1982, Tom Brigham, animador y programador del Laboratorio de Gráficos por Computador del Instituto Tecnológico de Nueva York, introdujo el morphing en la conferencia anual de gráficos por computador SIGGRAPH (<http://www.siggraph.org/>). El presentó un vídeo constituido por una secuencia de imágenes en la cual se transformaba una mujer en un lince. Sin embargo, su utilización comercial no fue inmediata. Sólo en 1987 el morphing fue utilizado por primera vez en el cine, en una escena para la película Willow realizada por la compañía Industrial Light and Magic ILM. Esta empresa también participó en varias películas donde se muestran escenas logradas con técnicas muy avanzadas de morphing tridimensional [35].

En 1990, otra compañía de gráficos por computador, Pacific Data Images PDI, aplicó el morphing en la creación de gráficos por computador, desarrollando su propia técnica de morphing y realizando con ella varios comerciales y vídeos musicales.

18.2 Técnicas de metamorfosis

Existen muchos métodos de metamorfosis, pero los más comúnmente utilizados son el algoritmo de warping por mallado, introducido por Doug Smythe de ILM y el de metamorfosis de campo, en inglés field morphing, introducido por Thaddeus Beier y Shawn Neely de PDI. Sin embargo, el método de morphing más simple consiste basicamente en la disolución de una imagen en otra mediante técnicas de composición. En esta técnica se realiza un cambio de colores de una imagen inicial a una final, mediante la interpolación de colores entre ellas.

18.2.1 Deformación por mallado

El algoritmo de warping por mallado relaciona algunas características entre una imagen origen y una destino a través de una malla no-uniforme. Primero, se crea una malla colocando líneas horizontales y verticales sobre la imagen origen y se crea otra correspondiente sobre la imagen destino. Para mover un vértice en la malla, el usuario lo selecciona con el ratón y lo arrastra hasta una nueva posición. Luego, las áreas de la malla de origen se transforman en las áreas de la malla de destino [37]. Así, se relacionan algunos puntos de la malla de origen, correspondiente a la imagen inicial con la malla de destino, correspondiente a la imagen final. Estos puntos determinan cómo se transforman las diversas partes de la imagen inicial en la final. La forma en que se realiza la transformación se mantiene aún bajo reserva y no ha sido divulgada.

18.2.2 Metamorfosis de campo

En el algoritmo de metamorfosis de campo se utilizan líneas, en lugar de vértices en una malla, para relacionar las características deseadas entre la imagen origen y la destino, tal como se ilustra en la figura 18.1. Cada línea en la imagen origen tiene una línea correspondiente en la imagen destino. Estas líneas determinan cómo se transforman los detalles de la imagen inicial en la final [2]. Esta técnica es ampliada a continuación.

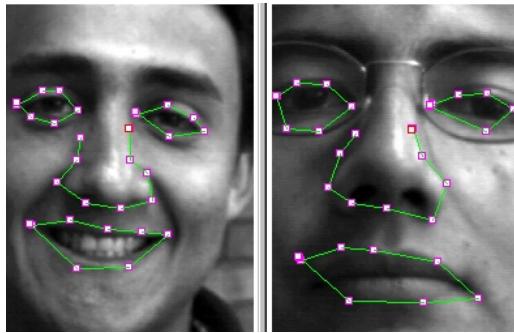


Figura 18.1: Líneas empleadas en técnica de metamorfosis de campo.

18.3 Metamorfosis de campo

18.3.1 Transformación de un par de líneas paralelas

Un par de líneas correspondientes en las imágenes fuente y destino, definen un mapeo inverso desde la coordenada del pixel x en la imagen destino, hasta la coordenada del pixel x' en la imagen fuente; tal que para una línea pq en la imagen destino y $p'q'$ en la imagen fuente donde la función Perpendicular() define el vector perpendicular \mathbf{Y} de la misma longitud que el vector de entrada. De aquí pueden establecerse los siguientes valores:

$$v = \frac{(x - p) \cdot \text{Perpendicular}(q - p)}{|q - p|}$$

$$x' = p' + u(q' - p') + \frac{v \cdot \text{Perpendicular}(q' - p')}{|q' - p'|}$$

$$u = \frac{(x - p)(q - p)}{|q - p|^2}$$

donde u es la posición sobre la línea y v es la distancia desde la línea al punto, tal como se aprecia en la figura 18.2. El valor u varía de 0 a 1 a medida que el pixel se desplaza de p a q y es menor que 0 o mayor que 1 fuera del rango. El valor de v es la distancia perpendicular en pixeles desde la línea [2].

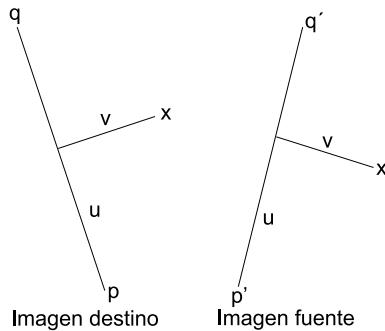


Figura 18.2: Relación entre las líneas en la imágenes fuente y destino en la técnica de metamorfosis de campo.

En la figura 18.2, el punto x' está en la localización correspondiente a la muestra en la imagen fuente para el pixel x en la imagen destino. x' está a una distancia v de la línea pq (distancia más corta desde la línea al pixel en la imagen origen y determinada por la perpendicular a la línea $p'q'$ que pasa por el punto x'), y a un valor proporcional u sobre la línea. El algoritmo transforma cada coordenada del pixel por medio de una serie de transformaciones geométricas (rotación, traslación y escalamiento) hasta transformar la imagen completa. Todos los pixeles sobre la línea en la imagen fuente son copiados sobre la línea en la imagen destino. Ya que la coordenada u es normalizada por la longitud de la línea mientras que la coordenada v no, la imagen es escalizada a lo largo de la dirección de las líneas por el radio de las longitudes de las mismas [2].

Múltiples pares de líneas indican transformaciones más complejas. Una ponderación de las transformaciones de las coordenadas para cada línea es desarrollado. Una posición x'_i es calculada para cada par de líneas. El desplazamiento $d_i = x'_i - x$ es la diferencia entre la localización del pixel en la imagen origen y la imagen destino y un peso promediado de estos desplazamientos es calculado. La distancia de x a la línea determina el peso. Este desplazamiento promedio es sumado a la posición actual del pixel para determinar la posición x' en la imagen fuente, por tanto:

$$\text{peso} = \left(\frac{\text{longitud}^p}{a + \text{dist}} \right)^b$$

La variable b determina cómo la fuerza relativa de diferentes líneas decae con la distancia. Si b es grande, los pixeles sólo son afectados por las líneas más cercanas. Si b es cero, cada pixel es afectado en igual forma por todas las líneas. Los valores de b más utilizados varían entre 0.5 y 2. El valor de p está por lo general en el rango de 0 a 1. Si p es 0, todas las líneas tienen el mismo peso relativo. Si p es 1, las líneas más largas tienen un mayor peso relativo que las más cortas.

A continuación se presenta el algoritmo para el cálculo de la transformación de múltiples líneas:

Para cada pixel x en la imagen destino:

$DSUM = 0.0$

$SumaPesos = 0$

Para cada línea p_iq_i :

Calcular u, v basado en p_iq_i

Calcular x'_i basado en u, v y $p'_iq'_i$

Calcular el desplazamiento $d_i = x'_i - x$ para esta línea

$dist$: distancia más corta entre x y p_iq_i

$$\text{peso} = \left(\frac{\text{longitud}^p}{a + \text{dist}} \right)^b$$

$$DSUM = DSUM + d_i * \text{peso}$$

$$SumaPesos = SumaPesos + \text{peso}$$

$$x' = x + \frac{DSUM}{SumaPesos}$$

$$\text{ImagenDestino}(x) = \text{ImagenFuente}(x')$$

Bibliografía

- [1] Bath University. http://www.bath.ac.uk/BUCS/Software/image_analysis/visilog/html/refguide. 1997.
- [2] Beier T. & Neely S. Feature-based image metamorphosis. Proceedings SIGGRAPH92. Computer Graphics Vol. 26 No. 2. Pág. 35-42. Julio 1992. También en <http://www.hammerhead.com/thad/morph.html>.
- [3] Besançon J.E. Vision par ordinateur en deux ou trois dimensions. Editions Eyrolles. 1988.
- [4] Bonilla, Gilberto. Interferencia estadística. Segunda edición. Ed. Trilla. Mexico D.F. 1991.
- [5] Bourennane E., Paindavoine M. & Truchelet F. Amélioration du filtre de Canny Deriche pour la détection des contours sous forme de rampe. Traitement du signal. Vol. 10. No. 4. Pág. 297-310. 1993.
- [6] Brown C.M. Inherent bias and noise in the Hough transform. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-5. No. 4. Pág. 493-505. Septiembre 1983.
- [7] Canny J. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence. Vol. PAMI-8. No. 6. Pág. 679-698. Noviembre 1986.
- [8] Carlson A.B. Sistemas de comunicación. Mc Graw-Hill. 1980.
- [9] Cocquerez J.P. & Philipp S. Analyse d'images: Filtrage et segmentation. Masson. 1995.
- [10] Collignon A. et al. 3D multy - modality medical image registration using feature space Clustering. First International conference CVRMed'95. Pág 195 - 204. Niza. Francia. Abril 1995.
- [11] Covell M. & Withgott M. Spanning the Gap between Motion Estimation and Morphing. Proceedings IEEE international conference in acoustic, speech & signal processing. Vol. 5. Pág.213-216. Nueva York. 1994. también en <http://www.interval.com/papers/icassp94/icassp94.html>. 16 Diciembre 1997.
- [12] Chavin S. et al. Reconocimiento de rasgos faciales utilizando modelos deformables. Grupo ITC. Universidad Complutense de Madrid. http://www.itc.ucm.es/PROYECTOS/SNAKE/p_snake.html. 1997.
- [13] Deriche R. Using Canny's criteria to derive a recursively implemented optimal edge detector. International Journal of computer vision. Vol. 1. No. 2. Pág. 167-187. 1987.
- [14] Didon J.P. & Langevin F. Registration of MR images : From 2D to 3D, using a projection based cross correlation method. Engineering in Medicine & Biology. 17th Annual Conference. IEEE CD ROM. ISBN 0-7803-2478-1. 1995.
- [15] Didon J.P. & Langevin F. Fast matching of MR images : validation of a projection based cross correlation method. Engineering in Medicine & Biology. 17th Annual Conference. IEEE. CD ROM. ISBN 0-7803-2478-1. 1995.
- [16] Duff T. & Porter T. Compositing Digital Images. Computer Graphics. Vol. 18. No. 3. Pág. 253-259. Julio 1984.
- [17] Faugeras O. Three-dimensional computer vision. A geometric viewpoint. The MIT Press. 1993.

- [18] Forero M.G & Arias E.A. Estudio del efecto de las máscaras de convolución en imágenes mediante el uso de la transformada de Fourier. Revista ingeniería e investigación. Universidad Nacional de Colombia. Aceptado para su publicación.
- [19] Gerig G., Kübler O., Kikinis R. & Jolesz F.A. Nonlinear anisotropic filtering of MRI data. IEEE Transactions on medical imaging. Vol. 11. No. 2. Pág. 221-232. Junio 1992.
- [20] Gonzalez R. C. & Woods R. E. Digital image processing. Addison-Wesley. 1992.
- [21] Haralick R.M. & Shapiro L.G. Survey: Image segmentation techniques. Computer vision, graphics and image processing. Vol 29. Pág. 100-132. 1983.
- [22] Haralick R. M., Sternberg S. R. & Zhuang X. Image analysis using mathematical morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-20. No. 4. Pág. 532-550. 1987.
- [23] Heikkilä J. & Silven O. Counting and tracking people from image sequences. University of Oulu. <http://www.ee.oulu.fi/~jth/monitor/>.
- [24] Hoogenboom R. A. The Karhunen-Loeve Transform. <http://www.wi.leidenuniv.nl/~rhoogenb/node27.html>. 1996.
- [25] Jacq J.J & Roux C. Automatic registration of 3D images using a simple genetic algorithm with a stochastic performance function. Engineering in Medicine & Biology. 17th Annual Conference. IEEE. CD ROM. ISBN 0-7803-2478-1. 1995.
- [26] Kapur J. N., Sahoo P. K. & Wong A. K. C. A new method for gray-level picture thresholding using the entropy of the histogram. Computer vision, graphics and image processing. Vol. 29. Pág. 273-285. 1985.
- [27] Kass M., Witkin A. & Terzopoulos D. Snakes : Active Contour Models. International Journal of Computer Vision. Pág. 321-331. 1988.
- [28] Lam L, Lee S. W. & Suen C.Y. Thinning methodologies - A comprehensive survey. IEEE Transactions on pattern analysis and machine intelligence. Vol. 14. No. 9. Pág. 869-885.
- [29] Lamer R. et al. Registration of reconstructed arteriovenous malformation (AVM) volume within a magnetic resonance (MR) volume. Engineering in Medicine & Biology. 17th Annual Conference. IEEE CD ROM. ISBN 0-7803-2478-1. 1995.
- [30] Lindley C. Practical image processing in C. John Wiley & Sons. 1991.
- [31] Liorzov A. Iniciación práctica a la estadística. Francisco Casanova De. Barcelona 1959.
- [32] Lifshitz L. M. & Pizer S. M. A multiresolution hierarchical approach to image segmentation based on intensity extrema. IEEE Transactions on pattern analysis and machine intelligence. Vol. 12. No. 6. Pág. 529-540. Junio 1990.
- [33] Marshall D. http://www.cm.cf.ac.uk/Dave/Vision_lecture. 1997.
- [34] Monga O. Des images aux surfaces. Université de Technologie de Compiegne. 1993. Reporte de Horaud R. & Monga O. Vision par ordinateur: Outils fondamentaux. Editions Hermès.
- [35] Morrison M. The Magic of image Processing, Sams Publishing. 1993.
- [36] Oliver D., Anderson S., McCord J., Gumas S. & Zigon B. Tricks of the graphics gurus. Sams Publishing. 1993.
- [37] Palma D.J. The Palma morphing paper. <http://www.norcosoft.com/ncs/palma.html>. 1994.
- [38] Parker J.R. Algorithms for image processing and computer vision. John Wiley & Sons. 1997.

- [39] Péria O. et al. Milimetric registration of SPECT and MR images of the brain without headholders. Engineering in Medicine & Biology. 17th Annual Conference. IEEE CD ROM. ISBN 0-7803-2478-1. Pág. 14-15. 1995.
- [40] Playboy. The lenna story. www.lenna.org. Noviembre 1972.
- [41] Pratt W. K. Digital image processing. 2a. edición. John Wiley & Sons. 1991.
- [42] Pérez N. Fundamentos del tratamiento de imágenes. Visión por computador. <http://bucanero.ugr.es/www.curso//ccordoba>. 3 Abril 1995.
- [43] Rakatobe R.H. et al. Echography and TI-201 spect cardiac image registration using elliptical models. Engineering in Medicine & Biology. 17th Annual Conference. IEEE. CD ROM. ISBN 0-7803-2478-1. 1995.
- [44] Russ, John C. The Image Processing Handbook. Segunda Edición. CRC Press. 1995.
- [45] Rudomin I. Modelos deformables. <http://journey.cem.itesm.mx/talkfis/node3.html>. Octubre 13 1995.
- [46] Sahoo P.K. et al. A survey of thresholding techniques. Computer vision, graphics and image processing. Vol. 41. Pág. 233-260. 1988.
- [47] Schreiner S. & Galloway Jr. R.L. A fast maximum intensity projection algorithm for generating magnetic resonance angiograms. IEEE Transactions on Biomedical Imaging. Vol. 2. No.1. Pág 50 - 57. Marzo 1993.
- [48] Vallino J. Oxford University Robotics Group. Oxford Snakes Library. <http://www.cs.rochester.edu/u/vallino/research/snakes/>.
- [49] Young D. University of Sussex. <http://www.cogs.susx.ac.uk/users/davidy/teachvision/vision4.html>. 1994.
- [50] Yoshimi B. Snakes, using active dynamic contours. Columbia University Robotics Group. http://www.cs.columbia.edu/robotics/projects/visual_control/bill-snakes.html.