

Segmentación

5.1. Extracción de bordes, esquinas y puntos de interés

Los puntos de borde, o simplemente bordes son pixeles alrededor de los cuales la imagen presenta una variación brusca en los niveles de gris.

El término borde se refiere a cadenas conectadas de puntos de borde, esto es fragmentos de contorno, esto no impide que en la imagen existan puntos aislados que presentan un alto contraste en los niveles de gris.

Esta detección de bordes es parte fundamental de la mayoría de sistemas de visión ya que el éxito de los siguientes niveles de procesamiento depende de la fiabilidad de las características de los bordes.

Para simplificar la explicación, nos enfocaremos en derivadas unidimensionales. En particular estamos interesados en el comportamiento de estas derivadas en áreas con niveles de gris constante (secciones planas), en el inicio y final de discontinuidades (escalones y rampas) y a lo largo de los niveles de gris de las rampas. Estos tipos de discontinuidades pueden ser usadas como modelos de puntos de ruido, líneas y bordes en una imagen. El comportamiento de las derivadas dentro y fuera de la imagen, también son de interés.

La derivada de una función digital (discreta en sus coordenadas espaciales y su amplitud) están definidas en términos de sus diferencias.

5.2. Concepto de la derivada en la extracción de bordes

Los bordes que se pueden observar en la Figura 5.1(a) suelen estar desdibujados como resultado del muestreo. Esto hace que se modelen como una rampa (b) (transición de oscuro a claro o viceversa) en lugar de hacerlo como un cambio brusco.

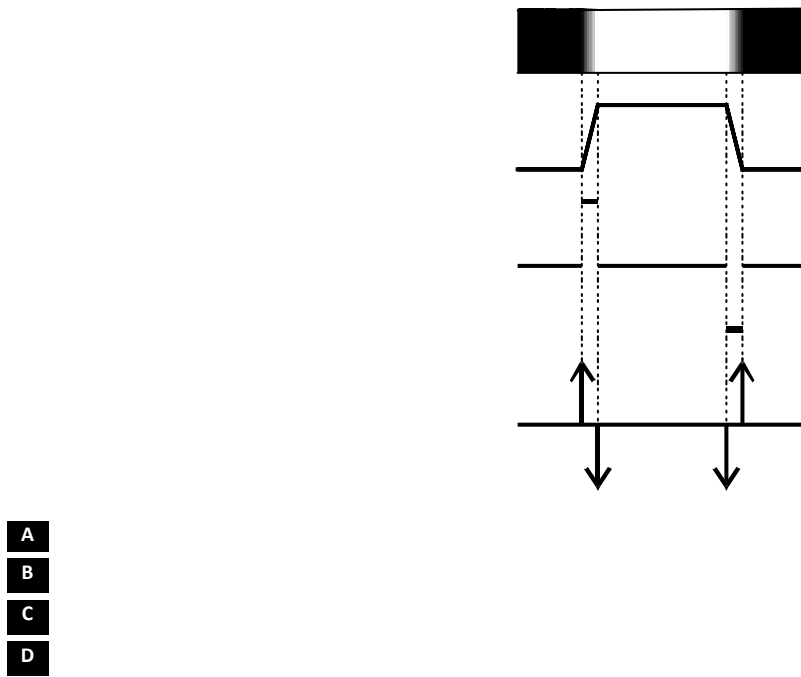


Figura 5.1: Concepto de primera y segunda derivada. **A.** Imagen con bordes; **B.** Perfil de intensidad de una línea horizontal (bordes desdibujados); **C.** Primera derivada de A y **D.** Segunda derivada de A.

Así, la primera derivada es:

- Cero en todas las regiones de intensidad constante y
- Tiene un valor constante en toda la transición de intensidad.

La segunda derivada, en cambio, es:

- Cero en todos los puntos, excepto al principio y final de una transición de intensidad.

Por tanto, un cambio brusco de intensidad se manifiesta como un cambio brusco en la primera derivada y presenta un paso por cero, es decir se produce un cambio de signo en su valor, en la segunda derivada.

Una definición básica de la derivada de primer orden de una función $f(x)$ unidimensional es la diferencia:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \quad (5.1)$$

Similarmente, definimos la derivada de segundo orden como la diferencia:

$$\frac{\partial^2 f}{\partial x} = f(x+1) + f(x-1) - 2f(x) \quad (5.2)$$

Usamos la notación de derivada parcial para cuando consideremos funciones bidimensionales (imágenes) $f(x, y)$ y así utilizar la misma notación.

La Figura 5.2 muestra el uso de las diferencias para obtener la primera y segunda derivada de una línea unidimensional horizontal de los niveles de gris de una imagen. En ella se pueden observar un punto de ruido (punto aislado), rampas, segmentos planos y un escalón. El comportamiento de la primera derivada se ve claro en la rampa que tiene un valor constante a lo largo de ella o en su segunda derivada que tiene un valor al principio y al final de ella.

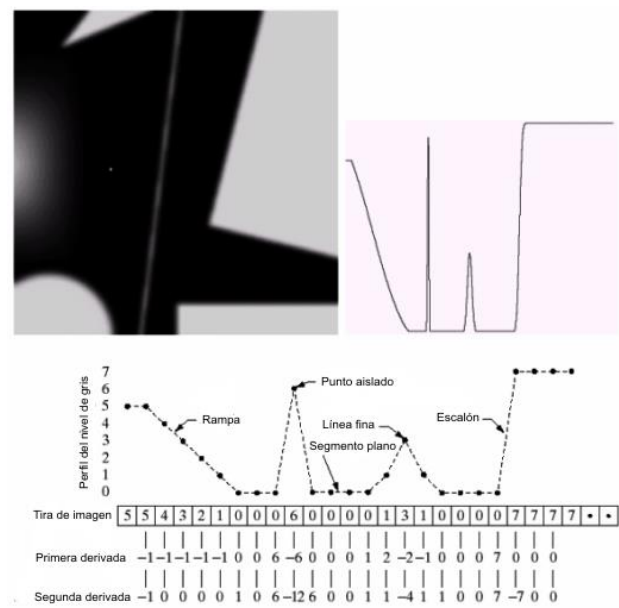


Figura 5.2: **A.** Imagen común. **B.** Perfil longitudinal de una línea unidimensional horizontal de niveles de gris del centro de la imagen que incluye un punto de ruido aislado. **C.** Simplificación del perfil (el punto está unido por líneas punteadas para simplificar la interpretación).

Al hacer un análisis de los resultados de la primera y segunda derivada, observamos que la primera derivada produce bordes más gruesos y la segunda derivada, en cambio produce bordes más finos. En los puntos aislados, la respuesta en y alrededor del punto es mayor en la segunda derivada que en la primera. Una segunda derivada, es por tanto, mucho más agresiva que una primera derivada en el realzado del cambio en el borde. Así podemos esperar que la segunda derivada realce detalles finos (incluyendo el ruido) mucho más que la primera derivada.

La observación anterior permite distinguir entre operadores de primera y segunda derivada.

5.3. Gradiente de una imagen

Como las imágenes son señales bidimensionales, la aplicación de la derivada se hace a través del gradiente en sus dos coordenadas espaciales.

El gradiente de una imagen $f(x, y)$ en el punto (x, y) se define como un vector bidimensional perpendicular al borde, dado por:

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix} \quad (5.3)$$

Donde G apunta en la dirección de variación máxima de f en el punto (x, y) por unidad de distancia con magnitud:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (5.4)$$

Y dirección:

$$\phi(x, y) = \tan^{-1} \frac{G_y}{G_x} \quad (5.5)$$

Es muy común aproximar la dirección de la magnitud del gradiente con los valores absolutos:

$$|G| \approx |G_x| + |G_y| \quad (5.6)$$

Esto se hace porque el valor del gradiente no llega a ser tan importante como la relación que hay entre diferentes valores. Es decir, se va a decidir si un punto es un borde según si la magnitud del gradiente supere o no un determinado umbral, entonces es solo cuestión de ajustar dicho umbral para que el resultado de la extracción de bordes sea el mismo tanto si se calcula la magnitud del gradiente mediante 5.4, o mediante 5.6, sin embargo, esta última ecuación es más fácil de implementar y particularmente cuando se realiza en hardware.

La relación de la función umbral (binaria) para determinar si un punto es borde o no, es:

$$g(x, y) = \begin{cases} 1 & \text{si } G[f(x, y)] > T \\ 0 & \text{si } G[f(x, y)] \leq T \end{cases} \quad (5.7)$$

Donde T es un valor de umbral no negativo. Sólo los píxeles de borde cuyo gradiente excedan del valor de T se consideran importantes. Así la ecuación anterior se puede ver como un procedimiento que extrae sólo aquellos píxeles caracterizados por transiciones de intensidad significativas (dependiendo de T).

Entonces la derivada en imágenes digitales (señales discretas bidimensionales) se calcula utilizando las diferencias de primer orden entre dos píxeles adyacentes, esto es:

$$G_x = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}; \quad G_y = \frac{f(y + \Delta y) - f(y - \Delta y)}{2\Delta y} \quad (5.8)$$

Esta es la forma más elemental de obtener el gradiente en un punto. La magnitud del gradiente puede tomar cualquier valor real y el ángulo también cualquier valor real entre 0° y 360° .

5.4. Operadores basados en la primera derivada

Como hemos indicado anteriormente, los valores de G_x y G_y pueden implementarse por convolución de la imagen con máscaras de 3×3 .

5.4.1. Operador Sobel

En la siguiente Figura 5.3 se muestran las máscaras conocidas como operadores Sobel:

$$\begin{bmatrix} Z_1 & Z_2 & Z_3 \\ Z_4 & Z_5 & Z_6 \\ Z_7 & Z_8 & Z_9 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



Figura 5.3: Máscaras denominadas operadores Sobel. **A.** Región de la imagen de dimensión 3×3 . **B.** Máscara usada para obtener G_x en el punto central de la región 3×3 . **C.** Máscara usada para obtener G_y en el mismo punto.

A partir de la Figura 5.3, las derivadas basadas en los operadores Sobel son:

$$\begin{aligned} G_x &= (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) \\ G_y &= (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) \end{aligned} \quad (5.9)$$

Donde los distintos valores de z en la región de la Figura 5.3a son los niveles de gris de los píxeles solapados por las máscaras en cualquier localización de la imagen. Para obtener los valores de las componentes del vector gradiente en el punto definido por el pixel central se la región se utilizan las últimas dos expresiones con lo que la magnitud y el ángulo se obtienen mediante 5.6 y 5.5 respectivamente, es decir obtenemos un valor del gradiente en dicho punto. Para obtener el siguiente valor, las máscaras se mueven a la siguiente posición del nuevo pixel y se repite el proceso, después de haber barrido todas las posibles posiciones, el resultado es una imagen gradiente. Una vez obtenido la magnitud del gradiente, se decide entonces si un determinado punto es de borde o no aplicando la ecuación de umbral, obteniendo así una imagen binaria como resultado final.

Ejemplo 5.1:

Supongamos la imagen de la Figura 5.4:

$$\begin{bmatrix} \circ 0 & \circ 8 & \circ 8 & 8 & 8 & 8 \\ \circ 1 & \bullet 8 & \circ 9 & 7 & 6 & 8 \\ \circ 2 & \circ 3 & \circ 4 & 8 & 8 & 8 \\ 2 & 2 & 2 & 8 & 8 & 8 \\ 2 & 2 & 2 & 2 & 8 & 8 \end{bmatrix}$$

Figura 5.4: Imagen para calcular el gradiente en G_x y G_y . El punto negro indica donde comienza la convolución de las máscaras de la Figura 5.2(b y c).

Se pretende calcular el gradiente en el pixel marcado con el punto negro, la región que interviene en la convolución está marcada con los puntos en blanco.

Calculamos el gradiente en las direcciones G_x y G_y :

$$G_x = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) = (8 + 2 \times 9 + 4) - (0 + 2 \times 1 + 2) = 30 - 4 = 26$$

$$G_y = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) = (2 + 2 \times 3 + 4) - (0 + 2 \times 8 + 8) = 12 - 24 = -12$$

Utilizamos 5.6 para obtener $|G|$:

$$|G| \approx |26| + |-12| = 38$$

Si fijamos $T=30$ tenemos:

$$g(x, y) = \begin{cases} 1 & \text{si } G[f(x, y)] > 30 \\ 0 & \text{si } G[f(x, y)] \leq 30 \end{cases}$$

Por lo que el pixel marcado con el punto negro sería un punto de borde, pero si marcamos a $T=48$, dicho punto ya no sería considerado de borde. A continuación desplazaríamos las máscaras a la siguiente posición (pixel con valor 9) situado a la derecha del que se acaba de procesar y así hasta terminar con toda la imagen.

5.4.2. Operador Prewitt

Este operador es similar al Sobel, diferenciándose en los coeficientes de las máscaras según la Figura 5.5.

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



Figura 5.5: Máscaras denominadas operadores Prewitt. **A.** Máscara usada para obtener G_x en el punto central de la región 3×3 . **B.** Máscara usada para obtener G_y en el mismo punto.

La magnitud y la dirección del gradiente se calculan de la misma manera que en el caso del operador Sobel.

5.4.3. Operador de Roberts

Este operador a diferencia de los anteriores, marca solamente los puntos de borde, sin informar sobre su orientación de éstos. Es un operador muy simple y trabaja bien con imágenes binarias. Para implementarlo existen dos formas:

1. Raíz cuadrada de la suma de las diferencias de los vecinos diagonales al cuadrado:

$$\sqrt{[f(x, y) - f(x-1, y-1)]^2 + [f(x, y-1) - f(x-1, y)]^2} \quad (5.10)$$

2. Suma de la magnitud de las diferencias de los vecinos diagonales

$$|f(x, y) - f(x-1, y-1)| + |f(x, y-1) - f(x-1, y)| \quad (5.11)$$

Siendo la más usada la segunda por su menor costo en tiempo computacional.

Las máscaras usadas según la ecuación 5.11 por el método de Roberts son:

$x+I, y-I$	$x+I, y$	$x+I, y+I$		
$x, y-I$	x, y	$x, y+I$	0	1
$x-I, y-I$	$x-I, y$	$x-I, y+I$	-1	0

A **B** **C**

Figura 5.6: Máscaras denominadas operadores Roberts. **A.** Región de la máscara usada por Roberts de dimensión 3x3 (parte sombreada). **B.** Máscara usada para obtener G_x . **C.** Máscara usada para obtener G_y .

Una desventaja del operador Roberts es que es muy sensible al ruido, ya que se usan muy pocos pixeles para aproximar el gradiente, además, la aproximación esta desplazada del punto (x, y) al punto $(x + \frac{1}{2}, y + \frac{1}{2})$.

5.4.4. Máscaras de Kirsch

Denominadas también como de brújula porque se definen considerando una máscara simple y rotándola en las 8 direcciones principales de la brújula (Figura 5.7).

<i>NO</i>	<i>N</i>	<i>NE</i>
<i>O</i>	<i>Imagen Original</i>	<i>E</i>
<i>SO</i>	<i>S</i>	<i>SE</i>

Figura 5.7: las 8 direcciones de la brújula para el operador Kirsch.

Las máscaras de definen en la Figura 5.8 como:

$$k_0 \equiv \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad k_1 \equiv \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad k_2 \equiv \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad k_3 \equiv \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$k_4 \equiv \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad k_5 \equiv \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad k_6 \equiv \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad k_7 \equiv \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

A	B	C	D
E	F	G	H

Figura 5.8: Máscaras de Kirsch en las 8 direcciones. **A.** 0° ; **B.** 45° ; **C.** 90° ; **D.** 135° ; **E.** 180° ; **F.** 225° ; **G.** 270° y **H.** 315° .

Para cada punto de la imagen se obtienen 8 valores, resultantes de la convolución con cada una de las máscaras, el valor del módulo del gradiente resulta ser el máximo de esos 8 valores, mientras que la dirección queda determinada por el ángulo asociado a la máscara que ha generado dicho valor máximo.

Por ejemplo, para el punto marcado en negro de la imagen de la Figura 5.4 tenemos:

$k_0 = 63$, $k_1 = 95$, $k_2 = 23$, $k_3 = -33$, $k_4 = -81$, $k_5 = -57$, $k_6 = -33$ y $k_7 = 23$, por lo que $\max\{k_i\} = k_1$, en cuyo caso $|G| = 95$ y la dirección del gradiente resulta ser 45° .

5.4.5. Máscaras de Robinson

Se usan de forma similar a las de Kirsch siendo ahora la máscara inicial la mostrada en la Figura 5.9.

$$r_0 \equiv \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figura 5.9: Máscara de Robinson generadora de las otras 7 máscaras restantes (el ángulo de $r_0 = 0^\circ$).

5.4.6. Máscaras de Frei-Chen

La lógica del método de Frei-Chen es ponderar los píxeles vecinos en función de la distancia que lo separa del pixel donde estamos evaluando. Esto hace que no haya direcciones privilegiadas en la detección de los bordes. El conjunto de 9 máscaras del método de Frei-Chen son:

$$\begin{aligned}
 f_1 &\equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} & f_2 &\equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} & f_3 &\equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \\
 f_4 &\equiv \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} & f_5 &\equiv \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} & f_6 &\equiv \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \\
 f_7 &\equiv \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} & f_8 &\equiv \frac{1}{6} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} & f_9 &\equiv \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
 \end{aligned}$$

Figura 5.10: Máscara de Frei-Chen.

Estas máscara forman un conjunto completo de vectores base, designado por $\{e\}$. Esto significa que se puede presentar cualquier subimagen de dimensión 3×3 como una suma promediada de las máscaras Frei-Chen. Los coeficientes de las sumas se encuentran proyectando una subimagen 3×3 en cada una de esas máscaras. Este proceso de proyección es similar al proceso de convolución en el sentido de que ambos superponen la máscara en la imagen, multiplican los términos coincidentes y suman los resultados.

Un ejemplo para aclarar el concepto de proyección consideremos las máscara de la Figura 5.11(a) y la porción de imagen de la Figura 5.11(b).

$$W = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} \quad Z = \begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$



Figura 5.11: **A.** Máscara de dimensión 3×3 . **B.** Subimagen de dimensión 3×3 .

Si se expresan (a) y (b) de la Figura 5.11 en forma de vectores tenemos las siguientes expresiones:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_9 \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_9 \end{bmatrix} \quad (5.12)$$

Con esta notación el resultado R de aplicar una máscara w a la subimagen z está dado por el producto escalar de los vectores w y z . Es decir:

$$R = \sum_{i=1}^9 w_i z_i = \|w\| \|z\| \cos \theta = w^t z \quad (5.13)$$

Donde t indica transposición de vectores, $\| \cdot \|$ indica el módulo del correspondiente vector y θ el ángulo entre dos vectores, que en definitiva se trata de una proyección.

Ejemplo 5.2:

Supongamos que tenemos la siguiente subimagen, I_S :

$$I_S = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Figura 5.12: Subimagen I_S de dimensión 3×3 .

Para proyectar esta subimagen en las máscaras Frei-Chen, solapamos la subimagen en cada una de las máscaras y obtenemos mediante la ecuación 5.13:

$$(I_S, f_1) = 0, \quad (I_S, f_2) = 0, \quad (I_S, f_3) = 0, \quad (I_S, f_4) = 0, \quad (I_S, f_5) = -1, \quad (I_S, f_6) = 0, \quad (I_S, f_7) = 0, \quad (I_S, f_8) = -1, \\ (I_S, f_9) = 2$$

Ahora, multiplicando los valores anteriores por el conjunto de vectores base se puede representar la subimagen por una suma promediada. Los vectores base son las máscaras de Frei-Chen y los pesos los valores de proyección. Para generar la imagen, tomar los pesos y multiplicarlos por cada máscara; luego sumar los correspondientes valores. Si seguimos los resultados del ejemplo 5.2 los únicos términos diferentes de cero corresponden a las máscaras f_5 , f_8 y f_9 , y se encuentra que,

$$I_s = (-1)\left(\frac{1}{2}\right) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + (-1)\left(\frac{1}{6}\right) \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} + (2)\left(\frac{1}{3}\right) \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Este ejemplo nos muestra como representar una subimagen como una suma promediada y ahora vamos a ver como se usan para detectar bordes. Las máscaras de Frei-Chen pueden agruparse en un conjunto de cuatro máscaras definiendo un subespacio de bordes con f_1 , f_2 , f_3 y f_4 , cuatro máscaras para un espacio de línea con f_5 , f_6 , f_7 y f_8 y una máscara para un subespacio medio, f_9 . Es decir, los subespacios de bordes y líneas sirven para detectar la presencia de puntos de borde en general y líneas respectivamente, el subespacio medio sirve para suavizar la imagen original, ya que es el operador de la media, promediando el entorno de vecindad.

Para hacer uso de las máscaras de Frei-Chen en la detección de bordes, se selecciona un subespacio de interés y se encuentra la proyección relativa de la imagen en dicho subespacio $\{e_s\}$, como sigue:

$$\cos \theta = \sqrt{\frac{M}{S}} \quad (5.14)$$

Donde: $M = \sum_{k \in \{e_s\}} (I_s, f_k)^2$ y $S = \sum_{k=1}^9 (I_s, f_k)^2$

La magnitud de la proyección viene determinada por M , de tal forma que dependiendo de su valor detectaríamos la presencia de un punto de borde.

5.5. Operadores basados en la segunda derivada

5.5.1. Operador Diferencia de Gaussianas (DoG)

5.5.1.1. Introducción e idea básica del operador

El método de la diferencia de gaussianas, también conocido como DoG (del inglés derivative of Gaussian), se basa en los estudios realizados por Marr y Hildreth.

Partiendo del hecho de que una imagen tiene componentes a muy diferentes frecuencias, pretender extraer toda la información con un único filtro es imposible. Por tanto, es necesario utilizar diferentes herramientas según los datos que sean de interés en cada caso.

El objetivo de este operador es la extracción de bordes de manera eficiente. Para conseguir buenos resultados se elimina mediante suavizado toda la información superflua. Así desaparecen las componentes de alta frecuencia, asociadas a pequeños detalles, mientras que las estructuras grandes permanecen inalteradas.

Experiencias con animales han puesto de manifiesto que éstos utilizan un método similar para la detección de bordes.

Un buen filtro de suavizado debe tener dos características:

1. **Localidad frecuencial:** Hay que hacer una limitación de altas frecuencias. Las altas frecuencias están asociadas a dos tipos de eventos: puntos y bordes. Un punto corresponde con una variación muy rápida de la imagen en una zona muy reducida del espacio (similar a una delta) mientras que para un borde la diferencia de nivel se extiende a lo largo de la dirección del mismo, teniendo un espectro más ancho en altas frecuencias, como veremos posteriormente.
2. **Localidad espacial:** Trabajar con zonas demasiado grandes de la imagen es muy costoso computacionalmente y además no permite trabajar con un margen amplio de frecuencias. Por otra parte, operar con tamaños muy reducidos impediría detectar los bordes.

El filtro gaussiano cumple ambos requisitos ya que mediante el valor de la varianza pueden seleccionarse rangos intermedios tanto en la frecuencia como en el espacio.

El suavizado por sí solo no permite la detección de bordes, sin embargo se pensó en la posibilidad de utilizar dos réplicas de la imagen original con diferentes suavizados para tener referencia de los bordes. El porqué la utilización de diferencias de imágenes desenfocadas es un método válido para la detección de bordes puede entenderse analizando una técnica de cuarto oscuro en fotografía convencional. Básicamente, la idea de esta técnica es la utilización de dos negativos de una misma imagen, uno es la imagen real y el otro una versión desenfocada de la misma (esto corresponde con la utilización de una campana gaussiana de desviación moderadamente alta). En la imagen desenfocada aparecen los bordes ensanchados y difuminados. Si se superponen ambos negativos y se hace un revelado del conjunto, la imagen obtenida sólo presenta cambios en las zonas donde los negativos no coinciden, como puede verse en la Figura 5.13.

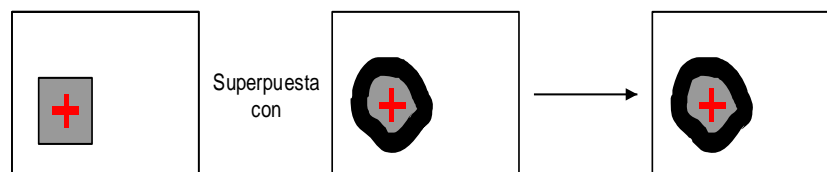


Figura 5.13: Superposición de negativos.

Si se hace esto, aplicando diferencias de imágenes, el resultado es que sólo se obtienen los bordes (Figura 5.14).

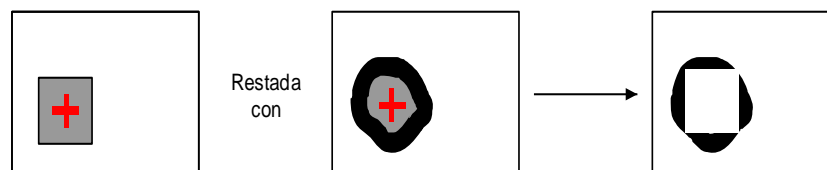


Figura 5.14: Sustracción de negativos.

Así queda demostrado que la diferencia de gaussianas es un método válido para la extracción de bordes en una imagen.

Una vez establecidas estas bases, el Método DoG se implementa aplicando a la imagen dos filtros gaussianos con diferentes desviaciones típicas y realizando la resta de los resultados. Ambos filtros tienen como objetivo promediar de forma ponderada el valor del pixel sobre

el que se está actuando con los valores de los píxeles colindantes. Campanas gaussianas anchas implican que se está teniendo en cuenta un mayor número de píxeles, entendiéndose con esto que el valor de sus pesos son suficientemente significativos en el ponderado. Frente a esto, la utilización de una desviación típica pequeña implica que se están teniendo en cuenta sólo los píxeles más próximos al píxel actual, ya que la campana es más estrecha.

Para una dimensión, la forma que adopta el operador DoG es el de la Figura 5.15.

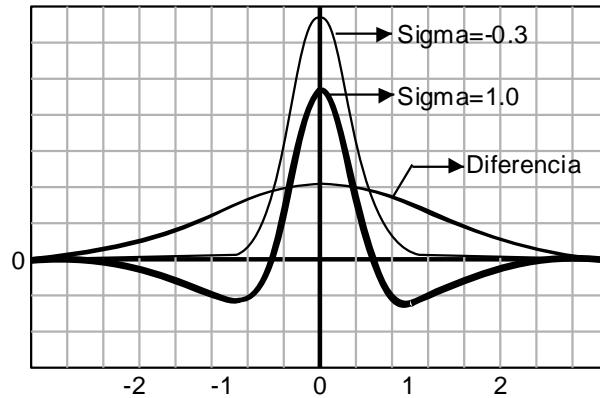


Figura 5.15: Forma que adopta el operador DoG.

Aunque para el tratamiento de imágenes la DoG se aplica en dos dimensiones, puede extraerse la información sobre su comportamiento de esta figura:

- Los pesos en los promediados de los píxeles vecinos quedan determinados por la amplitud de las gaussianas en cada uno de ellos.
- La utilización de la diferencia de ambas campanas provoca la aparición de cruces por cero y por tanto de cambios de signo. Ambos elementos están asociados con la existencia de un borde.
- Un borde implica un cambio moderado de nivel de intensidad. Al considerar para el promediado una región próxima al píxel, se va a producir una modificación respecto al valor real de intensidad en el borde. Del mismo modo, por estar trabajando con regiones de la imagen mayores que un píxel, se va a producir un ensanchado de los bordes. La combinación de ambos fenómenos puede entenderse como un difuminado de los bordes.

5.5.1.2. Fundamentos matemáticos

El suavizado de una imagen se va a realizar mediante la convolución de la misma con un filtro gaussiano.

En el caso continuo, la función gaussiana de desviación típica σ y media μ , tiene la expresión

$$y_{\sigma}(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2} \quad (5.15)$$

El valor de σ determina la anchura de la campana, mientras que la media μ indica el punto donde estará centrada dicha campana.

Para trabajar con imágenes es necesario extender este operador a dos dimensiones, de modo que pueda operarse sobre un plano. En este caso la forma que adopta el operador es:

$$y_{\sigma}(x, y) = \frac{1}{2\pi \cdot \sigma^2} e^{-\frac{1}{2} \left(\frac{x^2 + y^2}{\sigma^2} \right)} \quad (5.16)$$

Una propiedad muy útil de este operador es que es separable. Esto quiere decir que puede operarse de forma independiente por filas (x) y columnas (y), para posteriormente realizar el producto de ambos resultados y obtener la función global. De este modo:

$$G_{\sigma}(x, y) = g_1(x) \cdot g_2(y) \quad (5.17)$$

siendo:

$$g_1(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x}{\sigma} \right)^2} \quad \text{y} \quad g_2(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{y}{\sigma} \right)^2} \quad (5.18)$$

Puesto que las imágenes digitalizadas son de naturaleza discreta, se debe trabajar con la versión discreta del operador gaussiano (filtro binomial). En este caso el aspecto del operador es el de una matriz cuadrada y simétrica, cuyos coeficientes y tamaño viene determinado por el valor de σ . Un convenio que se suele adoptar para elegir el tamaño W de la matriz es:

$$W \geq 3 \cdot C \quad (5.19)$$

siendo: $C = 2\sigma\sqrt{2}$ el tamaño del lóbulo de la gaussiana.

El resultado de la convolución de una imagen ($f(i, j)$) con el filtro gaussiano discreto, definido por una matriz, es el siguiente:

$$F(i, j) = \frac{1}{2\pi \cdot \sigma^2} \cdot \sum_i \sum_j e^{-\frac{1}{2} \frac{k^2 + l^2}{\sigma^2}} \cdot f(i + k, j + l) \quad (5.20)$$

La aplicación de este filtro a toda la imagen implica N^2 operaciones, sin embargo el hecho de que el operador sea separable en filas y columnas, permite la reducción del cómputo total de operaciones de N^2 a $2N$, siendo por tanto más rápido.

El filtro DoG se implementa con la diferencia de los resultados de convolucionar la imagen con dos gaussianas como las anteriormente descritas. Las gaussianas se construyen con matrices del mismo tamaño (W), para poder establecer la diferencia entre ambas en igualdad de condiciones. Sin embargo, las desviaciones típicas de las dos gaussianas que definen las dos matrices son diferentes, y por tanto también lo serán los coeficientes de las mismas. En cualquier caso para W debe cumplirse la relación con respecto a la anchura del lóbulo

principal establecida anteriormente por 5.19. Puesto que la anchura del lóbulo principal es función de $\sigma (C = 2\sigma\sqrt{2})$, y ahora se aplica dos desviaciones típicas diferentes, se tienen dos valores diferentes para c , pero la desigualdad ha de ser cierta en los dos casos.

Se indica a continuación el resultado de aplicar el operador sobre un borde:

Se parte de una imagen formada por dos zonas de colores diferentes con un cambio abrupto entre ellas. Cuando el operador se va desplazando sobre una de las zonas de color uniforme (suficientemente lejos del borde), el resultado de la DoG dará cero (ya que el valor de los promediados con σ_1 y σ_2 es el mismo).

Conforme la matriz se va desplazando hacia el borde, va tomando valores de ambos lados de la discontinuidad. A un lado y a otro los colores son distintos, y también lo serán las amplitudes de sus píxeles. Por tanto, los resultados de los promediados son diferentes en función de: la cantidad de píxeles de cada color que se estén tomando para su cálculo y los pesos de las gaussianas. Así la diferencia de las gaussianas puede ser distinta de cero. El hecho de que tome un valor positivo, negativo o nulo, dependerá de la imagen y de las desviaciones con las que se esté trabajando. Como vemos, aparecerán cambios de signo y cruces por cero a un lado y a otro de la discontinuidad, puntos que el algoritmo detectará como bordes.

Las consideraciones que se ha hecho para una transición abrupta entre dos colores, son extensibles a un caso general con una mayor gama de colores dentro de la imagen.

5.5.1.2 Ventajas e Inconvenientes

Las ventajas son:

- Es un operador simétrico, es decir, es capaz de detectar bordes en todas las direcciones. Esto supone una ventaja frente a otros operadores como por ejemplo los basados en la primera derivada, que requieren matrices diferentes para direcciones diferentes.
- La realización de un suavizado sobre la imagen elimina los pequeños detalles, lo cual hace que el filtro sea menos sensible al ruido. De hecho este algoritmo surgió como solución al problema del ruido en el operador laplaciano.
- La diferencia de gaussianas da como resultado cambios de signo (o cruces por cero) allí donde hay un borde. La detección de cambios de signo es sencilla y por tanto también lo es la implementación del algoritmo. Así, es más sencillo que los métodos de primera derivada cuyos objetivos son buscar máximos.
- El hecho de que el operador sea separable y tenga carácter local hace que su costo computacional sea relativamente bajo.

Los inconvenientes son:

- El filtrado por convolución con una matriz de una determinada dimensión provoca un ensanchamiento de los bordes, así como un ligero desplazamiento de los mismos.
- El operador DoG puede marcar como bordes zonas que no lo son, como por ejemplo, zonas de variación rápida del color pertenecientes a un mismo objeto.

5.5.3. Operador Laplaciana de la Gaussiana (Log)

5.5.3.1. Detección de bordes con LoG

La teoría de detección de bordes que proponen Marr y Hildreth en 1979, expone que los cambios en una imagen, ya sea por brillo o por color, pueden ser detectados encontrando los ceros de $G(x, y) * I(x, y)$ para una imagen I y donde $G(x, y)$ es una distribución Gaussiana de dos dimensiones, es decir, debemos convolucionar la imagen con un filtro gaussiano para obtener una imagen de salida a la cual habría que buscarle sus cruces por cero.

Estamos hablando de imágenes, por lo tanto son operadores de dos dimensiones, de modo que la expresión del gradiente de la Gaussiana corresponde a:

$$G(r, c) = G(r)G(c) = \frac{1}{2\pi\sigma^2} e^{-\frac{(r-m_r)^2 + (c-m_c)^2}{2\sigma^2}} \quad (5.21)$$

Este proceso de detección de contornos puede resumirse en dos pasos fundamentales:

Paso 1. Convolución de la imagen con el operador LoG

Paso 2. Localización de los cruces por cero

Nota: Existe un paso posterior que es "Verificación del alineamiento y la orientación de una segmento local de cruces por ceros" que debería utilizarse si el filtro con el que se convoluciona no tuviera las características de la LoG. Este método no utiliza la Laplaciana de Gaussiana por casualidad, se ha escogido porque es independiente de la dirección, dentro de los operadores derivativos de segundo orden, el único que es independiente de la dirección es el operador de Laplace por tanto nos libramos de implementarlo como tercer paso gracias a que lo resolvemos ya con este operador.

5.5.3.2. Paso primero. Convolución de la imagen con el operador LoG

Como se ha expuesto, se trata de convolucionar la imagen con el operador LoG (del inglés Laplacian of Gaussian) que es la Laplaciana de la Gaussiana de una señal, al utilizarla se están filtrando las variaciones de alta frecuencia en la imagen debido a que la LoG es un filtro paso bajo, consiguiendo la combinación de filtrado y detección de bordes al encontrar los ceros de la función. Aprovechando las propiedades de linealidad de estas operaciones, se asume la convolución de la LoG con la función.

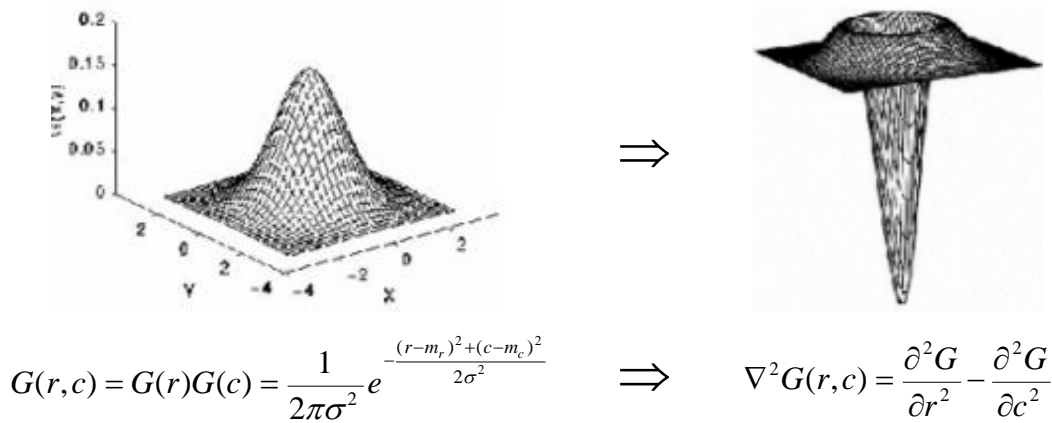


Figura 5.16: Obtención de la LoG a partir de la Gaussiana.

El hecho de trabajar con la LoG significa que tenemos una segunda derivada de una función con lo que el ruido asociado es considerable, esto se verá más adelante con la elección de las diferentes máscaras, según las cuales quedará filtrado este ruido en mayor o menor medida, la gran ventaja de utilizar este operador es que tendremos el mismo campo escalar como resultado, es decir, la LoG de una imagen es otra imagen de modo que en la implementación del programa nos es posible representar la imagen convolucionada igual que la entrada al programa.

La convolución de la imagen con el filtro LoG se realiza mediante una máscara o kernel que opera en los pixels recorriendo toda la matriz punto a punto, de modo que como la operación resultante depende del pixel central y sus vecinos, estamos hablando de un operador local.

Dependiendo del tamaño de los kernels con los que operemos sobre la imagen tendremos unos resultados con una apariencia muy distinta, esto se debe a que estamos implementando un filtro de LoG con mayor resolución cuando el kernel es de mayor tamaño lo que le permite una detección más fiable de los bordes pero también asume un mayor número de variaciones en la señal por lo que el ruido se incrementa de manera notable en la salida al filtrar con máscaras mayores.

La Figura 5.17 muestra tres kernels implementables.

-2	-4	-2
-4	24	-4
-2	-4	-2

-1	-3	-4	-3	-1
-3	-1	6	-1	-3
-4	6	24	6	-4
-3	-1	6	-1	-3
-1	-3	-4	-3	-1

0	-2	-3	-4	-3	-2	0
-2	-4	-3	-1	-3	-4	-2
-3	-3	6	13	6	-3	-3
-4	-1	13	24	13	-1	-4
-3	-3	6	13	6	-3	-3
-2	-4	-3	-1	-3	-4	-2
0	-2	-3	-4	-3	-2	0

A **B** **C**

Figura 5.17: 3 tipos de kernels para la LoG. **A.** 3x3; **B.** 5x5 y **C.** 7x7.

5.5.3.1. Paso segundo. Localización de los cruces por cero

En la Figura 5.18 tenemos lo que significa la detección de bordes a partir de una señal, dado que se trata de obtener el paso por ceros de su segunda derivada, como se puede apreciar es en este punto donde tendremos un borde, así que con la salida de la función implementada habremos obtenido una representación fiable de los bordes de la imagen a partir de esos puntos donde se anula la salida de la convolución con el LoG en el paso anterior. Lo más conveniente para un resultado satisfactorio con un operador local de filtrado es que la pendiente de la derivada (la 2a. en este caso) sea muy abrupta para que se haya definido con más precisión el punto donde se encuentra el cero.

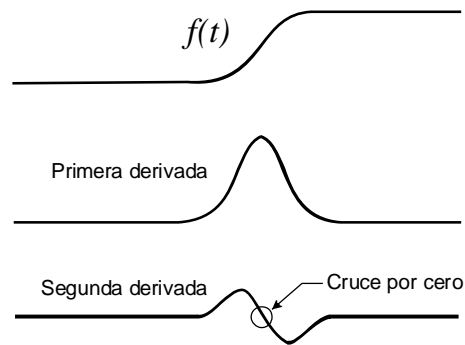


Figura 5.18: Cruce por cero en la segunda derivada.

Una importante mejora que se puede aplicar en estos algoritmos de filtrado para obtener bordes es el ajuste de los mismos mediante alguna técnica de detección de umbrales, de modo que nos saltaríamos cualquier variación en la imagen que no suponga un salto considerable en la segunda derivada, hay que tener cuidado con la selección de estos umbrales porque no solo podríamos saltarnos los cambios de tonalidad dentro de una figura, sino algunas otras variaciones que puedan caracterizar la imagen como uno de sus bordes, con un umbral muy bajo los bordes salen representados con mayor anchura de la que tienen en la realidad.

5.6. Extracción de Regiones

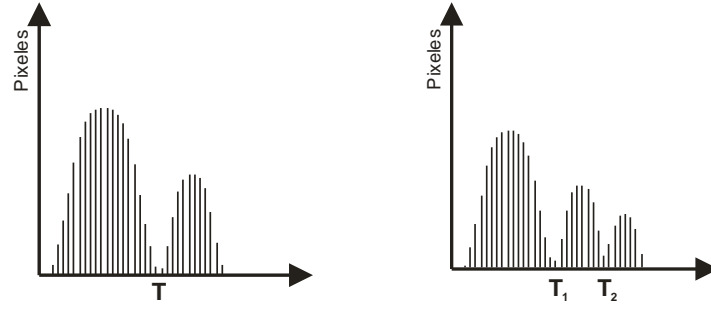
Las unidades de las imágenes son los píxeles y las técnicas de filtrado de imágenes transforman píxeles en píxeles. Las únicas propiedades de un píxel son su posición y su nivel de gris.

En las imágenes aparecen ciertas áreas o zonas caracterizadas por el hecho de que constituyen agrupaciones de píxeles conectados entre sí, pero además de su conexión, dichos píxeles presentan propiedades o características comunes.

5.6.1. Binarización mediante detección del umbral

Supongamos que el histograma de intensidad mostrado en la Figura 5.19(A) corresponde a una imagen $f(x, y)$ compuesta por un objeto claro sobre un fondo oscuro, teniendo los píxeles del objeto y del entorno intensidades agrupadas en dos tonos dominantes. Una forma

de extraer los objetos del entorno es seleccionar un nivel T que separe los dos tonos de intensidad. De esta forma, un pixel (x, y) para el cual $f(x, y) > T$ será un pixel del objeto; en caso que suceda lo contrario, el punto será del fondo.



A **B**

Figura 5.19: Histogramas de imágenes $f(x, y)$. **A.** Bimodal y **B.** Trimodal.

En la Figura 5.19(B) se muestra un caso ligeramente más general del uso de esta técnica. Ahora el histograma está caracterizado por tres modos dominantes (dos tipos de objetos sobre un fondo oscuro). El proceso será similar: se clasifica un punto como perteneciente a una de las clases de objetos si $T_1 < f(x, y) \leq T_2$, a la otra clase de objeto si $f(x, y) > T_2$ y al fondo si $f(x, y) \leq T_1$.

Los procedimientos que utilizan este tipo de umbrales (multinivel) suelen ser menos fiables que aquellos que usan el umbral simple debido a las dificultades de establecer umbrales múltiples que aislen efectivamente regiones de interés.

Basándonos en los conceptos anteriores, podemos considerar la fijación de un umbral como una operación que implica pruebas con respecto a una función T de la forma:

$$T = T[x, y, p(x, y), f(x, y)] \quad (5.22)$$

Donde $f(x, y)$ es la intensidad en el punto (x, y) y $p(x, y)$ es alguna propiedad local del punto; por ejemplo, la intensidad media de un entorno de vecindad cerrado en (x, y) . Se crea, entonces, una imagen binaria $g(x, y)$ definida:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases} \quad (5.23)$$

Examinando $g(x, y)$ se ve que los pixeles a los que se les asigne valor 1 corresponden al objeto, mientras que los que corresponden al fondo tienen el valor de 0

Cuando T depende únicamente de $f(x, y)$, el umbral se llama global. Si T depende tanto de $f(x, y)$ como de $p(x, y)$, se llama al umbral local y si T depende de las coordenadas espaciales x e y , se le llama umbral dinámico.

5.6.2. Selección del umbral óptimo

Como hemos mencionado es muy difícil determinar cuál es el umbral óptimo para poder llevar a cabo una binarización adecuada, entonces debemos desarrollar un método para seleccionar el umbral más apropiado.

A menudo es posible considerar un histograma como formado por la suma de funciones de densidad de probabilidad. En el caso del histograma bimodal, la función global que aproxima el histograma viene dada por:

$$p(z) = P_1 p_1(z) + P_2 p_2(z) \quad (5.24)$$

Donde z es una variable aleatoria que representa la intensidad, $p_1(z)$ y $p_2(z)$ son las funciones de densidad de probabilidad, P_1 y P_2 son las prioridades a priori. Estas dos últimas son las probabilidades de ocurrencia de los dos tipos de niveles de intensidad de la imagen. Por ejemplo consideremos una imagen cuyo histograma se muestra en la Figura 5.20.

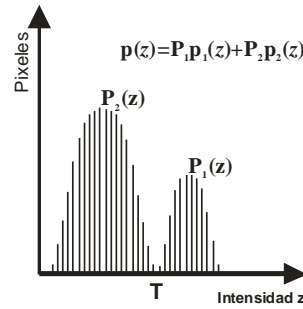


Figura 5.20: Suma de funciones de densidad.

El histograma global se puede aproximar por la suma de sus dos funciones de densidad de probabilidad. Si se sabe que los píxeles iluminados representan el objeto y se sabe también que estos ocupan el 20% del área de la imagen, entonces $P_1 = 0.2$. Debiéndose cumplir que, $P_1 + P_2 = 1$ el cual nos dice, en este caso, que el 80% restante son píxeles de fondo.

Formemos 2 funciones de densidad de z :

$$d_1(z) = P_1 p_1(z) \text{ y } d_2(z) = P_2 p_2(z) \quad (5.25)$$

De la teoría de decisiones, se sabe que el error medio de clasificar equivocadamente un píxel de objeto como de fondo, o viceversa, se minimiza usando la siguiente regla: dado un píxel con valor de intensidad z , se sustituye este valor de z en las ecuaciones anteriores. Clasificamos el píxel como píxel de objeto si $d_1(z) > d_2(z)$ o como píxel de fondo y caso contrario. El valor óptimo viene dado entonces por el valor de z para el cual $d_1(z) = d_2(z)$. Esto es, haciendo, $z = T$ en las mismas últimas ecuaciones y deducimos que el umbral óptimo satisface la ecuación

$$P_1 d_1(z) = P_2 d_2(z) \quad (5.26)$$

Si suponemos que $p_1(z)$ y $p_2(z)$ son funciones de densidad gaussianas y por tanto conocidas, esto es:

$$p_1(z) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(z-m_1)^2}{2\sigma_1^2}} \quad (5.27)$$

$$p_2(z) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(z-m_2)^2}{2\sigma_2^2}} \quad (5.28)$$

Donde: m_1, m_2 son los valores medios de nivel de brillo (intensidad) y σ_1, σ_2 son las desviaciones estándar respecto a la media

Haciendo $z = T$ en estas expresiones, sustituyendo en 5.26 y haciendo una simplificación se obtiene una ecuación de segundo grado en T :

$$AT^2 + BT + C = 0 \quad (5.29)$$

Donde:

$$\begin{aligned} A &= \sigma_1^2 - \sigma_2^2 \\ B &= 2(m_1\sigma_2^2 - m_2\sigma_1^2) \\ C &= \sigma_1^2 m_2^2 - \sigma_2^2 m_1^2 + 2\sigma_1^2 \sigma_2^2 \ln \frac{\sigma_2 P_1}{\sigma_1 P_2} \end{aligned} \quad (5.30)$$

La existencia de dos posibles soluciones indica que se pueden requerir dos valores de umbral para obtener una solución óptima.

Si las desviaciones estándar son iguales $\sigma_1 = \sigma_2 = \sigma$, será suficiente un umbral único:

$$T = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln \frac{P_2}{P_1} \quad (5.31)$$

Si $\sigma = 0$ o $P_1 = P_2$ el nivel óptimo es el valor medio de las medias. La primera condición nos dice que tanto la intensidad del objeto como la del entorno son constantes en toda la imagen. La segunda condición significa que los pixeles del objeto y los del entorno tienen igual probabilidad de ocurrencia, condición que se cumple siempre que el número de pixeles del objeto sea igual al número de pixeles del entorno.

El procedimiento desarrollado anteriormente es aplicable a la selección de múltiples niveles. Supóngase que pudiéramos modelar un histograma multinivel como la suma de n funciones de densidad de probabilidad de forma que:

$$p(z) = P_1 p_1(z) + \dots + P_n p_n(z) \quad (5.32)$$

La aplicación de los métodos de umbral se podría ver como una clasificación de los píxeles entre n categorías posibles. La regla de decisión del error mínimo estaría basada ahora en n funciones de densidad de la forma:

$$d_i(z) = P_i p_i(z), \text{ con } i = 1, 2, \dots, n \quad (5.33)$$

Un píxel dado con una intensidad z se asigna a la k -ésima categoría si $d_k(z) > d_j(z)$ siendo $j = 1, 2, \dots, n$; con $j \neq k$. Como antes, el umbral óptimo entre la categoría k y la categoría j , llamado T_{kj} se obtiene resolviendo la ecuación:

$$P_k p_k(T_{kj}) = P_j p_j(T_{kj}) \quad (5.34)$$

5.7. Descriptores morfológicos (como se caracterizan las regiones)

5.7.1 Descripción del código de Freeman

Arreglo numerado de 8 direcciones: horizontal derecha (0) y a la izquierda (4), vertical hacia arriba (2) y hacia abajo (6), y las 2 diagonales (1, 3, 5 y 7). Esta roseta de 8 direcciones se conoce como código de Freeman (Figura 5.21).

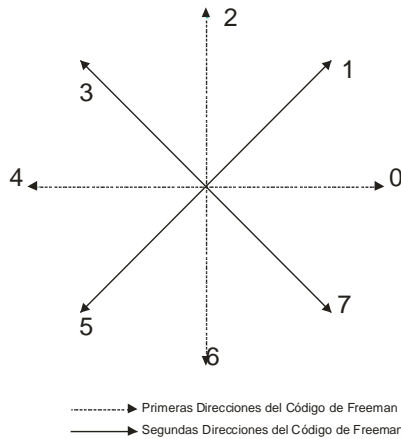


Figura 5.21: Roseta de 8 direcciones del código de Freeman.

En la sección 3.9.1 definimos la relación de vecindad de un píxel. Ahora esta relación de primeros y segundos próximos vecinos a un píxel p con coordenadas (i, j) los vamos a asociar al código de freeman. Los primeros próximos vecinos a este píxel p son aquellos con coordenadas $(i+1, j)$, $(i, j-1)$, $(i-1, j)$, $(i, j+1)$, que se encuentran en las direcciones (0, 2, 4, 6) del código de Freeman (línea punteada de la Figura 5.20). Los segundos próximos vecinos de p son aquellos localizados en las diagonales principales especificadas por las coordenadas: $(i+1, j-1)$, $(i-1, j-1)$, $(i-1, j+1)$, $(i+1, j+1)$ que se encuentran en las direcciones (1, 3, 5, 7) del código de Freeman (línea continua de la Figura 5.20).

Con la descripción de estas direcciones podemos entonces conocer los descriptores morfológicos, ya que estos se conectarán mediante tales direcciones.

5.7.2. Descriptores

5.7.2.1. Área

Número de píxeles que conforman la región (bajo cierta conectividad) multiplicado por la escala de observación.

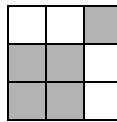


Figura 5.22: Imagen para ejemplificar el área según cierta conectividad.

Si el pixel = 10cm. \Rightarrow Bajo: Conectividad 4, una región de 40 y otra de 10 y Conectividad 8, una región de 50.

5.7.2.2. Perímetro

La distancia acumulada que hay entre los píxeles que componen el perímetro de la región (bajo cierta conectividad) multiplicado por la escala de observación.

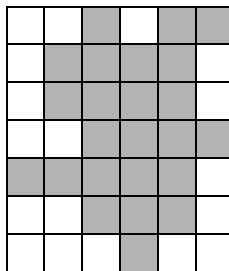


Figura 5.23: Imagen para ejemplificar el perímetro según cierta conectividad.

Bajo: Conectividad 4, $32 \times 10\text{cm} = 320$ (lados de los píxeles)

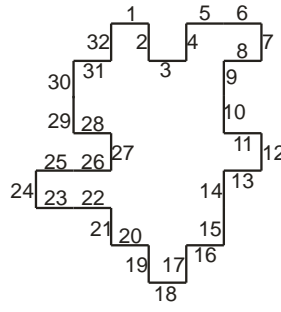
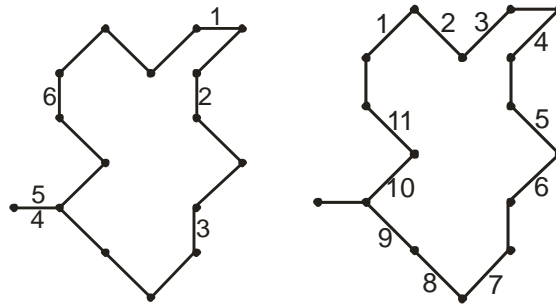


Figura 5.24: Cálculo del perímetro bajo conectividad 4.

Bajo: Conectividad 8, $6 + 11\sqrt{2} = 21.56 \times 10 \text{ cm} = 215.56$ (ver Figura 5.25, medición a partir de los centros del pixel). Para las segundas direcciones del Código de Freeman se multiplica por $\sqrt{2}$ (Figura 5.25(b)).



A **B**

Figura 5.25: Cálculo del perímetro bajo conectividad 8. **A.** Primeras Direcciones; **B.** Segundas Direcciones.

5.7.2.3. La Compacidad o Compactación (cuanto se parece una región a un círculo)

$$C = \frac{P^2}{4\pi A} \quad (\text{Bajo cierta conectividad}) \quad (5.35)$$

Donde: P es el perímetro y A es área

Para conectividad 4 de la Figura 5.24: $C = \frac{32^2}{4\pi(24)} = \frac{1024}{301.60} = 3.4$

Para conectividad 8 de la Figura 5.25: $C = \frac{21.56^2}{4\pi(24)} = \frac{464.83}{301.60} = 1.54$

Originalmente aplicada a espacios continuos y adaptada a espacios discretos.

$$C_d = \frac{0.6122A}{\left[\frac{1}{N_o} \sum_i d_{rr}^{\min} \right]^2} \quad (5.36)$$

Donde d_{rr}^{\min} es la distancia mínima de borde a borde del objeto de una línea que pasa por el pixel i-ésimo de este y N_o es el número de pixeles de la región digital

Distancias mínimas de la Figura 5.23:

$$d_{rr}^{\min} \Rightarrow \begin{array}{|c|c|c|c|c|c|} \hline & & 1 & & 2 & 1 \\ \hline & 2 & 4 & 4 & 4 & \\ \hline & 2 & 4 & 4 & 4 & \\ \hline & & 4 & 4 & 4 & 1 \\ \hline 1 & 1 & 5 & 5 & 5 & \\ \hline & & 3 & 3 & 3 & \\ \hline & & & 1 & & \\ \hline \end{array} \Rightarrow \sum_i d_{rr}^{\min} = 72$$

Figura 5.26: Obtención de las distancias mínimas de borde a borde de una línea que pasa por el pixel i-ésimo de la Figura 5.23.

$$C_d = \frac{0.6122(21.56)}{\left[\frac{1}{24}(72) \right]^2} = \frac{13.199}{9} = 1.4665 \quad (\text{bajo conectividad 8})$$

Una variante es la siguiente formula:

$$C_d' = \frac{A}{\bar{d}_{rr}^2} \quad (5.37)$$

Donde:

$$\bar{d}_{rr} = \left[(\bar{d}_{rr}^{\max}) + (\bar{d}_{rr}^{\min}) \right] / 2 \quad (5.38)$$

y:

\bar{d}_{rr}^{\max} : Promedio de las distancias máximas de borde a borde.

\bar{d}_{rr}^{\min} : Promedio de las distancias mínimas de borde a borde.

Distancias mínimas y máximas de la Figura 5.23:

$$\begin{pmatrix} d_{rr}^{\max} & d_{rr}^{\min} \end{pmatrix}$$

		1,6	2,6	1,2
	2,4	4,6	4,6	4,6
	2,4	4,6	4,6	4,6
		4,6	4,6	4,6
1,5	1,5	5,6	5,6	5,6
		3,6	3,6	3,6
		1,6		

Figura 5.27: Obtención de las distancias mínimas y máximas de borde a borde de una línea que pasa por el píxel i-ésimo de la Figura 5.23.

$$\bar{d}_{rr} = \left\lfloor \frac{(\bar{d}_{rr}^{\max}) + (\bar{d}_{rr}^{\min})}{2} \right\rfloor = \left\lfloor \frac{5.5 + 3}{2} \right\rfloor = 4.25 \text{ (bajo conectividad 8)}$$

$$C_d' = \frac{A}{\bar{d}_{rr}^2} = \frac{21.56}{4.25^2} = \frac{21.56}{18.0625} = 1.1936 \text{ (bajo conectividad 8)}$$

Un círculo en una malla de 6X6 con esta última formula arroja un resultado de $C_d' = 1.10$

5.7.2.4. Rugosidad del Perímetro

La distribución de curvaturas del perímetro de un objeto. La curvatura en el píxel i-ésimo P_i del perímetro se define como el cambio angular, de acuerdo al Código de Freeman, dado entre los vectores definidos por los pares de píxeles: $\{P_{i-1}, P_i\}, \{P_i, P_{i+1}\}$

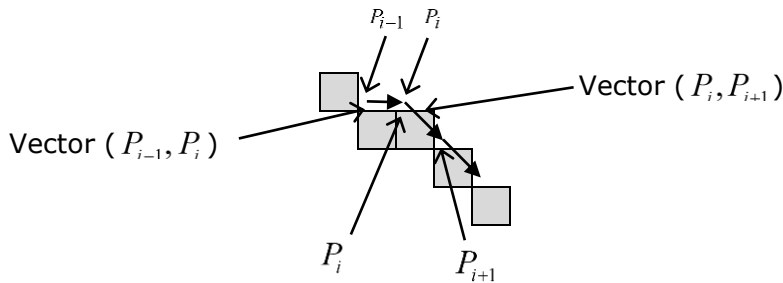
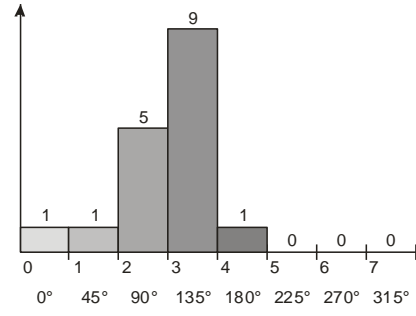


Figura 5.28: Pares de vectores que definen el cambio angular.

Se realiza un histograma de los cambios angulares del perímetro del objeto de acuerdo al Código de Freeman.

Para conectividad 8 de la Figura 5.23:

Dir. C.F.	0	1	2	3	4	5	6	7
Ang (int)	0°	45°	90°	135°	180°	225°	270°	315°
	1	1	5	9	1	0	0	0



A **B**

Figura 5.29: Cambios angulares del perímetro del objeto de acuerdo al Código de Freeman. **A.** Suma de cambios angulares; **B.** Histograma de A.

5.7.2.5. La Excentricidad (medida de elongación de un objeto)

Se define como:

$$\xi = \frac{MAX[d_{rr}^{max}]}{MAX[d_{rr}^{min}]} \quad (5.39)$$

Para conectividad 8 de la Fig. 5.23:

$$\xi = \frac{MAX[d_{rr}^{max}]}{MAX[d_{rr}^{min}]} = \frac{6}{5} = 1.2$$

5.7.2.6. Diversidad o Estructura del Paisaje

Supóngase que una imagen ha sido segmentada en un conjunto de regiones: $\{R_1, R_2, \dots, R_c\}$. Si P_i y A_i son el perímetro y el área de la región R_i , si N_i es el número de regiones adyacentes a R_i y si N_c es el número total de regiones se define un parámetro de textura expresado como:

$$E_s^i = 2 \frac{\ln(P_m^i)}{\ln(A_i)} \quad (5.40)$$

Donde P_m^i es el perímetro modificado, dado por:

$$P_m^i = P_i + \left[\frac{2N_i(A_i - 1)}{N_c - 1} \right] \quad (5.41)$$

La diversidad del paisaje se expresa como el promedio de los valores del parámetro de estructura para todas las regiones.

$$E_s = \frac{1}{N_c} \sum E_s^i \quad (5.42)$$

5.7.2.7. Textura

Matriz de co-ocurrencia –una medida de la función de densidad conjunta-. Medida de la heterogeneidad (heterogéneo.- Compuesto de partes de diversa naturaleza. heterogeneidad.- Mezcla de partes de diversa naturaleza en un todo) de una región de la imagen.

Si tenemos la imagen de la Figura 5.30 y se nos da una $\beta \rightarrow 0$ tenemos que $|\beta|=1$, siendo la primera dirección del Código de Freeman.

1	4	5	5	4
3	3	2	4	2
4	3	1	3	1
4	5	4	4	5
1	5	2	3	3

Figura 5.30: Imagen para obtener la matriz de co-ocurrencia.

Por la primera dirección del Código de Freeman los pares de píxeles adyacentes se leerán de derecha a izquierda, teniendo además de la Figura 5.30 un rango de valores de 1 a 5. Tenemos la matriz de co-ocurrencia mostrada en la Figura 5.31

	1	2	3	4	5
1			1/20	1/20	1/20
2			1/20	1/20	
3	2/20	1/20	2/20		
4		1/20	1/20	1/20	3/20
5		1/20		2/20	1/20

Figura 5.31: Cambios angulares del perímetro del objeto de acuerdo al Código de Freeman. **A.** Suma de cambios angulares; **B.** Histograma de A.

El denominador de la matriz de co-ocurrencia es el número total de pares de píxeles obtenidos de la imagen 5.30 que para este caso es 20.