

Capítulo 4

Segmentación de imágenes

4.1 Introducción

El análisis de imágenes comprende todos los métodos y técnicas que se utilizan para extraer información de una imagen. El primer paso para ello lo constituye la segmentación de imágenes que se ocupa de descomponer una imagen en sus partes constituyentes, es decir, los objetos de interés y el fondo, basándose en ciertas características locales que nos permiten distinguir un objeto del fondo y objetos entre si.

La mayoría de las imágenes están constituidas por regiones o zonas que tienen características homogéneas (nivel de gris, textura, momentos, etc.). Generalmente estas regiones corresponden a objetos de la imagen. La segmentación de una imagen consiste en la división o partición de la imagen en varias zonas o regiones homogéneas y disjuntas a partir de su contorno, su conectividad, o en términos de un conjunto de características de los píxeles de la imagen que permitan discriminar unas regiones de otras. Los tonos de gris, la textura, los momentos, la magnitud del gradiente, la dirección de los bordes, las modas de los tonos de gris en ventanas 3x3, 7x7 y 15x15, etc., son características a utilizar para la segmentación. Distinguiremos entre **segmentación completa**, cuando las regiones disjuntas corresponden directamente a objetos de la imagen y **segmentación parcial**, cuando las regiones no se corresponden directamente con objetos de la imagen. Para conseguir la segmentación completa se necesita un nivel superior de conocimiento que utiliza un conocimiento específico del dominio de la escena. Este conocimiento de nivel superior puede ser, por ejemplo, que los objetos de la imagen corresponden a caracteres numéricos o letras de un alfabeto.

La operación de segmentación trata de distinguir si un píxel pertenece, o no, a un objeto de interés y, por lo tanto, produce una imagen binaria. Todavía no hay una teoría unificada de la segmentación de imágenes, solamente disponemos de un conjunto de algoritmos.

Los algoritmos de segmentación de imágenes monocromáticas se basan en alguna de las tres propiedades siguientes:

- a) *Discontinuidad* en los tonos de gris de los píxeles de un entorno, que permite detectar puntos aislados, líneas y aristas (bordes).
- b) *Similaridad* en los tonos de gris de los píxeles de un entorno, que permite construir regiones por división y fusión, por crecimiento o por umbralización.
- c) *Conectividad* de los píxeles desempeña un papel importante en la segmentación de imágenes. Recordemos que una región D se dice **conexa** o conectada si para cada

par de píxeles de la región existe un camino formado por píxeles de D que los conecta. Un **camino** de píxeles es una secuencia de píxeles **adyacentes** (que pertenecen a su entorno inmediato).

Los métodos de segmentación se pueden agrupar en cuatro clases diferentes:

a) Métodos basados en píxeles, que a su vez pueden ser:

- locales (basadas en las propiedades de los píxeles y su entorno)
- globales (basadas en la información global obtenida, por ejemplo, con el histograma de la imagen).

b) Métodos basados en bordes.

c) Métodos basados en regiones, que utilizan las nociones de homogeneidad y proximidad geométrica, como las **técnicas de crecimiento, fusión o división**.

d) Métodos basados en modelos.

Antes de pasar a estudiar cada uno de estos modelos vamos a ver técnicas para la detección de puntos, rectas, bordes y contornos, como herramientas previas.

4.1.1 Detección de puntos

Un *punto aislado* de una imagen tiene un tono de gris que difiere significativamente de los tonos de gris de sus píxeles vecinos, es decir, de los ocho píxeles de su entorno 3×3. Una máscara para detectar un punto aislado es la siguiente:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Aplicando esta máscara al píxel (i,j) obtenemos:

$$g(i,j) = -f(i-1,j+1) - f(i-1,j) - f(i+1,j+1) - f(i-1,j) + 8f(i,j) - f(i+1,j) - f(i-1,j-1) - f(i,j-1) - f(i+1,j-1)$$

Diremos que el píxel (i,j) es un punto aislado si

$$|g(i,j)| > T$$

donde T es el valor umbral fijado por el decisor. Dicho valor depende de la aplicación que estemos realizando.

Sin embargo, esta máscara puede detectar, como puntos aislados, píxeles que forman parte de un borde. Por ello, es más conveniente utilizar el filtro no lineal siguiente:

$$R(i, j) = \min_{\substack{(r,s) \in N_8(i,j) \\ (r,s) \neq (i,j)}} |f(r, s) - f(i, j)|$$

Diremos que el píxel (i, j) es un punto aislado si

$$|R(i, j)| > T.$$

4.1.2 Detección de líneas

Una *línea* es una secuencia de píxeles en la que dos píxeles consecutivos están conectados, es decir, son vecinos en un entorno 3×3 de alguno de ellos. Cada píxel se puede conectar con alguno de sus 8 píxeles vecinos, y por lo tanto, vamos a tener sólo 4 direcciones (tramos lineales) posibles: horizontal, vertical, de 45° y de -45° . En consecuencia, cualquier curva digital va a venir dada por una línea constituida sólo por tramos lineales de cualquiera de estos cuatro tipos. Así, para la detección de líneas podemos utilizar una máscara 3×3 que se irá moviendo por los píxeles de la imagen. Los píxeles que forman parte de una línea tendrán respuestas extremas. Por ejemplo, para detectar una raya o segmento horizontal (línea recta), cada píxel del segmento tiene un tono de gris que difiere del tono de gris de los píxeles vecinos superiores e inferiores. Por ello, podemos utilizar la siguiente máscara para detectar una línea recta horizontal:

$$M_1 = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

Para detectar rayas verticales, de 45° y de -45° , podemos utilizar, respectivamente, las máscaras siguientes:

$$M_2 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}, \quad M_3 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}, \quad \text{y} \quad M_4 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Obsérvese que la dirección preferida de cada máscara está ponderada con un coeficiente mayor que las demás direcciones posibles.

Aplicaremos las cuatro máscaras anteriores a todos los píxeles de la imagen. Sea $g_k(i, j)$ el resultado de aplicar la máscara M_k centrada en el píxel (i, j) . Fijado un valor umbral $T > 0$, diremos que el píxel (i, j) constituye una parte horizontal de una línea si

$$|g_1(i, j)| > |g_k(i, j)|, \quad \forall k \neq 1, \text{ y} \\ |g_1(i, j)| > T.$$

4.1.3 Detección de bordes

Un *borde* o *arista* es la frontera entre dos regiones cuyos tonos de gris difieren significativamente o tienen propiedades diferentes, como ocurre en el caso de texturas. Si queremos detectar los bordes tenemos que poner énfasis en los cambios bruscos de los niveles de gris de píxeles vecinos y suprimir aquellas áreas con valores de gris constantes. Los operadores derivada serán útiles para realizar dichas tareas.

Las técnicas de segmentación basadas en los bordes son adecuadas cuando las regiones son suficientemente homogéneas de manera que la transición entre regiones se pueda realizar en base a los tonos de gris. En caso contrario es más adecuado utilizar las técnicas de segmentación basadas en regiones.

Un *borde local* (*eje o arista local*), es un píxel cuyo nivel de gris difiere significativamente del nivel de gris de algunos píxeles de su entorno. Es decir, hay diferencia de contraste local. Ello se debe esencialmente a dos situaciones.

- a) El píxel forma parte del borde entre dos regiones diferentes de la imagen (cada región tiene cierta homogeneidad en sus niveles de gris, con respecto a algún criterio de homogeneidad).
- b) El píxel forma parte de un arco muy fino sobre un fondo de diferente nivel de gris.

Para detectar los bordes comenzaremos detectando los bordes locales. Los bordes locales se detectan midiendo la tasa de cambio de los tonos de gris de su entorno. Vamos a utilizar, para ello, el operador gradiente (como operador de primer orden) o el operador Laplaciano (como operador de segundo orden).

Como habíamos visto, el *operador diferencia* en la dirección del eje de abscisas viene dado por la expresión:

$${}^+D_x(m,n) = f(m+1,n) - f(m,n) \quad (\text{diferencia adelantada}),$$

que corresponde a la máscara $M_{1 \times 2} = [-1 \quad 1]$, es decir,

$${}^+D_x(m,n) = f(m,n) * M_{1 \times 2};$$

También se puede definir como:

$${}^-D_x(m,n) = f(m,n) - f(m-1,n) \quad (\text{diferencia atrasada}),$$

que corresponde con la máscara $M_{1 \times 2} = [-1 \quad 1]$.

La imagen ${}^+D_x(m,n)$ (o ${}^-D_x(m,n)$) detecta los bordes verticales de la imagen, pues destaca aquellos píxeles cuyo nivel de gris difiere sustancialmente de sus correspondientes píxeles de la izquierda.

Análogamente, se definen los dos operadores derivada en la dirección del eje de ordenadas, como:

$${}^+\mathbf{D}_y(m,n) = f(m,n+1) - f(m,n)$$

$$\mathbf{D}_y(m,n) = f(m,n) - f(m,n-1)$$

La imagen ${}^+\mathbf{D}_y(m,n)$ (o $\mathbf{D}_y(m,n)$) detecta los bordes horizontales.

Sin embargo, buscamos operadores que detecten los bordes sea cual sea su dirección, es decir, un **detector de bordes isotrópico**. Para ello, vamos a utilizar el operador gradiente:

$${}^+\mathbf{D}(m,n) = \begin{pmatrix} {}^+\mathbf{D}_x(m,n) \\ {}^+\mathbf{D}_y(m,n) \end{pmatrix}$$

La magnitud de un vector (gradiente) es invariante frente a rotaciones del sistema de coordenadas. La magnitud del vector gradiente es:

$$|{}^+\mathbf{D}(m,n)| = \left({}^+\mathbf{D}_x^2(m,n) + {}^+\mathbf{D}_y^2(m,n) \right)^{1/2}$$

Este operador no es lineal y requiere más tiempo de cómputo. Por ello, también se suele utilizar el operador:

$$|{}^+\mathbf{D}(m,n)| = |{}^+\mathbf{D}_x(m,n)| + |{}^+\mathbf{D}_y(m,n)|$$

pero este operador no es isotrópico y es más sensible a los bordes diagonales que a los bordes verticales u horizontales.

Los métodos basados en el gradiente son más adecuados cuando la transición en los niveles de gris de los píxeles del entorno es brusca, similar a una función escalón (paso). Sin embargo, estos operadores son pocos sensibles a los cambios graduales en los niveles de gris, que corresponden a funciones tipo rampa. En estos casos es más adecuado aplicar operadores basados en las derivadas de segundo orden, como, por ejemplo, el operador Laplaciano, definido a partir de la laplaciana de una función:

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

En nuestro caso, donde las imágenes son digitales (discretas), este operador se obtiene aplicando consecutivamente los operadores gradiente en cada dirección, es decir,

$$\nabla^2 f(m,n) = {}^+\mathbf{D}_x[{}^-\mathbf{D}_x(f(m,n))] + {}^+\mathbf{D}_y[{}^-\mathbf{D}_y(f(m,n))]$$

$$= {}^+\mathbf{D}_x[f(m,n) - f(m-1,n)] + {}^+\mathbf{D}_y[f(m,n) - f(m,n-1)]$$

$$= [f(m+1,n) - f(m,n) - (f(m,n) - f(m-1,n))] + [f(m,n+1) - f(m,n) - (f(m,n) - f(m,n-1))]$$

$$= f(m+1,n) + f(m,n+1) + f(m-1,n) + f(m,n-1) - 4f(m,n)$$

que se puede escribir como la convolución de f con la plantilla

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Es decir, $\nabla^2 f = f * \mathbf{M}$, aunque se suele utilizar $-\nabla^2 f$ en lugar de $\nabla^2 f$, y por tanto, $-\mathbf{M}$ en lugar de \mathbf{M} . Obsérvese que ${}^+\mathbf{D}_x[{}^-\mathbf{D}_x(f(m,n))]$ es la convolución de f con la plantilla

$$\begin{pmatrix} 1 & -2 & 1 \end{pmatrix}$$

y que ${}^+\mathbf{D}_y[{}^-\mathbf{D}_y(f(m,n))]$ es la convolución de f con la plantilla

$$\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$$

Es decir,

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

que nos dice que la imagen final es la suma de una imagen de bordes verticales con una imagen de bordes horizontales.

Por otra parte, como

$$\nabla^2 f(m,n) = f(m+1,n) + f(m,n+1) + f(m-1,n) + f(m,n-1) - 4f(m,n)$$

también se puede poner de la forma

$$= [f(m+1,n) - f(m,n)] + [f(m,n+1) - f(m,n)] + [f(m-1,n) - f(m,n)] + [f(m,n-1) - f(m,n)]$$

se puede interpretar como un filtro de primer orden que viene dado por la suma de las diferencias del tono de gris del píxel en cuestión con los cuatro píxeles de su entorno N_5 .

Asimismo, como

$$-\frac{1}{4} \nabla^2 f(m,n) = f(m,n) - \frac{1}{4} [f(m+1,n) + f(m,n+1) + f(m-1,n) + f(m,n-1)]$$

resulta que el filtro laplaciano viene a ser el complementario del filtro de paso-baja cuya plantilla es

$$\frac{1}{4} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Es decir, si le restamos a la imagen original la imagen borrosa obtenida mediante dicho filtro de paso-baja se obtiene la imagen de bordes producida por el filtro Laplaciano.

Análogamente, como

$$-\frac{1}{5}\nabla^2 f(m,n) = f(m,n) - \frac{1}{5}[f(m,n) + f(m+1,n) + f(m,n+1) + f(m-1,n) + f(m,n-1)]$$

se tiene la siguiente relación entre plantillas:

$$\frac{1}{5}\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} - \frac{1}{5}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Por lo tanto, el filtro Laplaciano se puede interpretar también como el complementario del filtro de paso-baja cuya plantilla es

$$\frac{1}{5}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

El filtro de Laplace no es isotrópico (es isotrópico sólo para pequeñas frecuencias) como se desprende de su función de transferencia (transformada de Fourier),

$$H(\omega_1, \omega_2) = e^{i\omega_1} + e^{-i\omega_1} - 4 + e^{i\omega_2} + e^{-i\omega_2} = 2\cos(\omega_1) + 2\cos(\omega_2) - 4$$

cuya representación (curvas de nivel) se muestra en la figura 1.

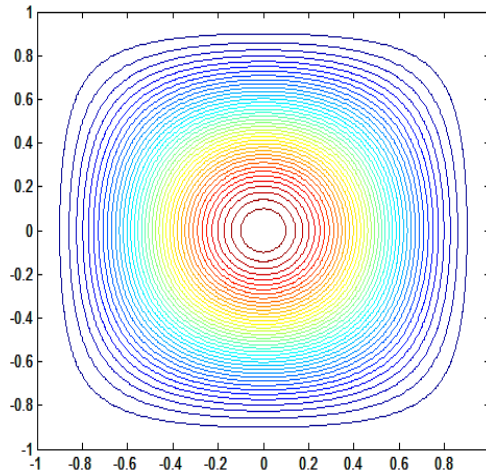


Figura 1. Curvas de nivel de la función de transferencia Laplaciana

La plantilla Laplaciana no se aplica directamente, pues al ser una diferencia de diferencias es muy sensible al ruido y, además, suele producir bordes desdoblados como puede verse en la figura 2 correspondiente a una imagen de granos de arroz.

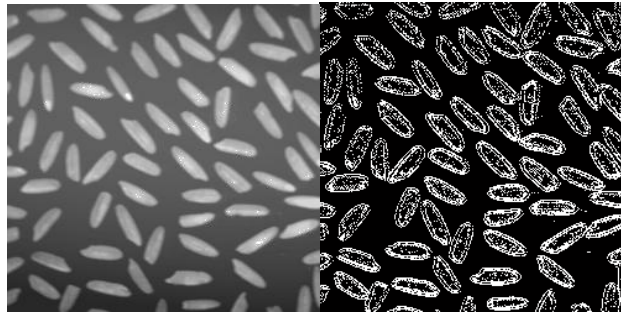


Figura 2. Detección de bordes mediante el filtro de Laplace y el operador umbral.

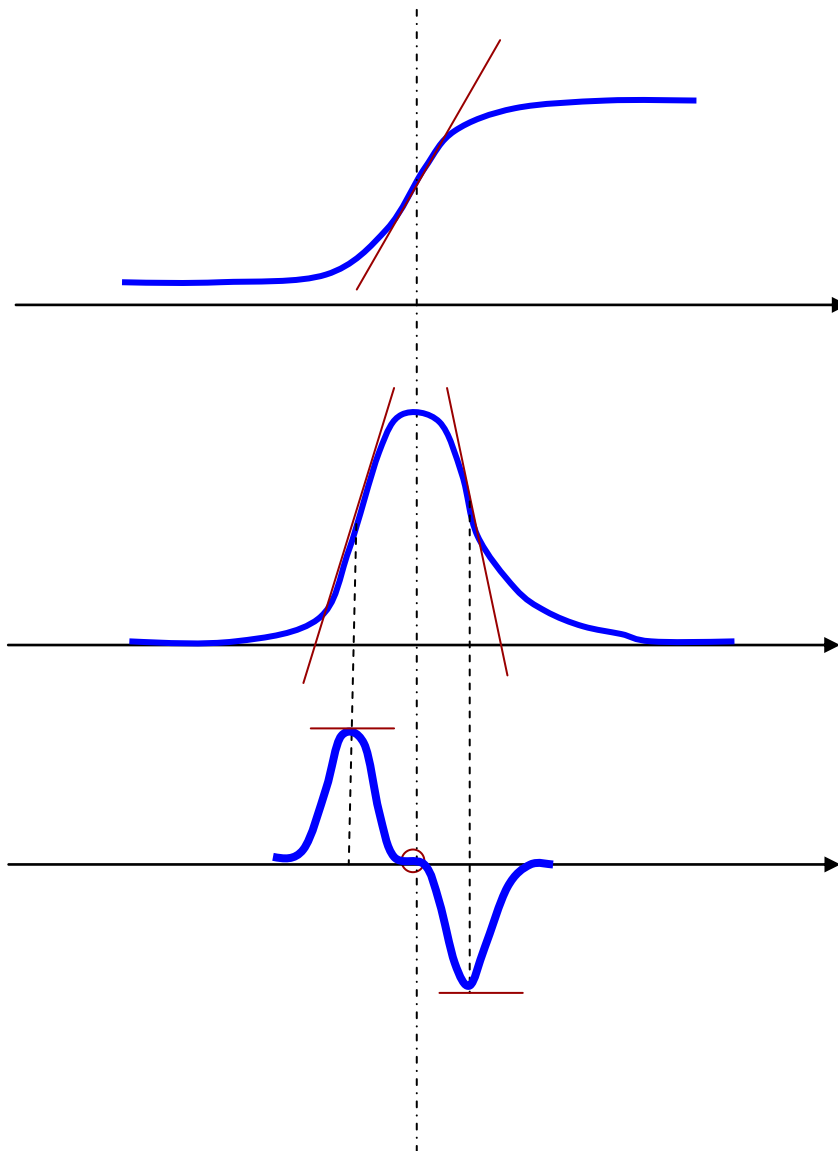


Figura 3. Derivadas sucesivas de una función rampa suavizada.

El desdoblamiento de borde se puede evitar si no aplicamos el operador umbral directamente sobre la imagen de bordes. Como se muestra en la figura 3, la derivada segunda de una función rampa suavizada, que puede representar el borde de un objeto, es una función con un máximo y un mínimo en lugar de una función con un máximo, como ocurre con la derivada primera. Por ello, para determinar los bordes debemos de encontrar los puntos de cruce con el origen en la imagen filtrada (puntos de cambios de signo). Es decir, las transiciones de valores positivos a negativos y viceversa.

Asimismo, para reducir su sensibilidad al ruido es conveniente filtrar previamente la imagen con un filtro de paso bajo, como puede ser el filtro Gaussiano:

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

Mediante el parámetro $\sigma > 0$ se controla el grado de suavización (borrosidad) deseado, y a continuación se le aplica el filtro de Laplace. El grado de suavización es proporcional a σ , conforme mayor es σ mayor borrosidad presentará la imagen filtrada. Obsérvese que aplicar el filtro de Laplace a la imagen suavizada

$$f^*(x, y) = \sum_r \sum_s f(r, s) h(x - r, y - s)$$

es equivalente a aplicar a la imagen $f(x, y)$ directamente el filtro

$$\nabla^2 h(x, y) = \left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right) e^{-(x^2+y^2)/2\sigma^2}$$

pues

$$\nabla^2 f^*(x, y) = \sum_r \sum_s f(r, s) \nabla^2 h(x - r, y - s).$$

Este filtro se conoce con el nombre de **Laplaciano de la Gaussiana**, y se suele representar por *LoG*. Las respuestas obtenidas con este filtro son positivas, negativas o nulas. La posición de los bordes locales viene determinada por los píxeles cuya respuesta cruza el cero. En la figura 4 se muestra la imagen de bordes obtenida utilizando el filtro Laplaciano de la Gaussiana a la imagen de granos de arroz.

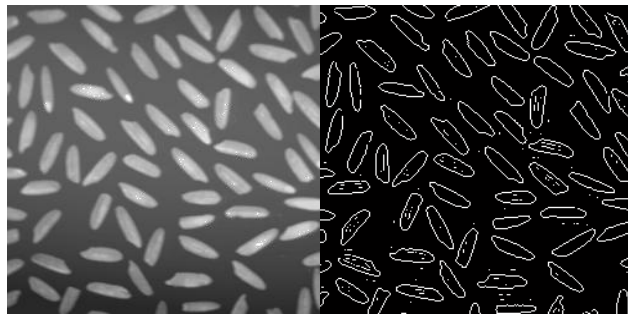


Figura 4. Detección de bordes mediante el filtro Laplaciano de la Gaussiana.

Otra forma de ver los bordes utilizando el filtro Laplaciano consiste en representar en negro los valores negativos de la imagen filtrada y en blanco los valores positivos. Así se obtiene una imagen binaria cuyos contornos vendrán dados aproximadamente por los puntos de la imagen que cruzan el cero (cambian de signo), es decir, los bordes de la imagen original.

Una aproximación al filtro anterior es el filtro *DoG* (**diferencia de gaussianas**). Viene dado por la expresión siguiente:

$$h(x, y) = e^{-(x^2+y^2)/2\sigma_1^2} - e^{-(x^2+y^2)/2\sigma_2^2},$$

siendo $\sigma_1 < \sigma_2$. Dicho filtro es de paso de banda.

A continuación vamos a estudiar el **método de Canny** para la detección de bordes. Canny (1986) propuso una metodología más complicada para la detección de bordes basada en tres criterios:

- El criterio de **detección** expresa el hecho de evitar la eliminación de bordes importantes y no suministrar bordes espurios.
- El criterio de **localización** establece que la distancia entre la posición real y la localizada del borde se debe minimizar.
- El criterio de **una respuesta** que integre las respuestas múltiples correspondientes a un único borde.

Algoritmo de Canny

1. Se suaviza la imagen $f(m, n)$ con un filtro gaussiano $G(m, n)$, con parámetro de escala σ ,

$$g(m, n) = e^{-\frac{m^2+n^2}{2\sigma^2}}$$

2. Para cada píxel de la imagen se estiman las magnitudes del gradiente

$$g(m, n) = \sqrt{D_x^2(m, n) + D_y^2(m, n)}$$

y las direcciones de los bordes locales

$$\alpha(m, n) = \tan^{-1}\left(\frac{D_x(m, n)}{D_y(m, n)}\right)$$

3. Se determinan los máximos locales (crestas) de las magnitudes del gradiente.

Comentario: Cuando los bordes son del tipo escalón, como en la figura 1(a), o rampa, la convolución con una gaussiana presenta los bordes de forma suavizada, como se muestra en la figura 5(b), y el borde local corresponde al punto de máxima pendiente, es decir, al valor máximo de su derivada. Por lo tanto, la ecuación anterior nos determina dicho máximo.

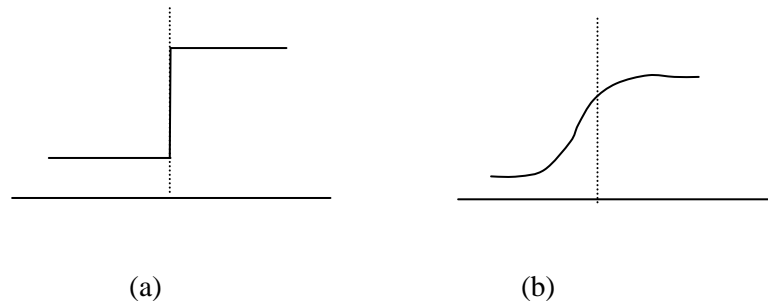


Figura 5. (a) Borde del tipo escalón. (b) Borde suavizado por convolución con la gaussiana.

4. **Supresión no maximal:** Se eliminan los bordes espurios mediante el operador umbral con *histéresis*. Es decir, se seleccionan dos valores umbrales, uno superior y otro inferior. Los píxeles cuyo valor de la magnitud del gradiente está por encima del umbral superior se consideran como partes de bordes y se les llama bordes fuertes, mientras que los que tienen una respuesta por debajo del umbral inferior no se consideran bordes. Los que están entre ambos valores se les llama bordes débiles
5. **Enlace de bordes:** El algoritmo incorpora como bordes aquellos bordes débiles que están conectados a bordes fuertes, es decir, que están dentro de un entorno 3×3 .

Se repiten los pasos anteriores para valores ascendentes del parámetro de escala σ .

6. Utilizando la información disponible acerca de los bordes para diferentes escalas se aplica la **síntesis de características** propuesta por Canny. Consiste en marcar primero todos los bordes detectados con el valor más pequeño del parámetro de escala y a partir de ellos se predicen los bordes que se detectarían con un operador para un valor grande de σ (bordes sintetizados). Entonces la respuesta sintetizada de bordes se compara con la respuesta de ella para σ grande. Se marca un borde adicional sólo si su respuesta es más fuerte que la predicha.

En la figura 6 se muestra la imagen de bordes obtenida aplicando el filtro de Canny.

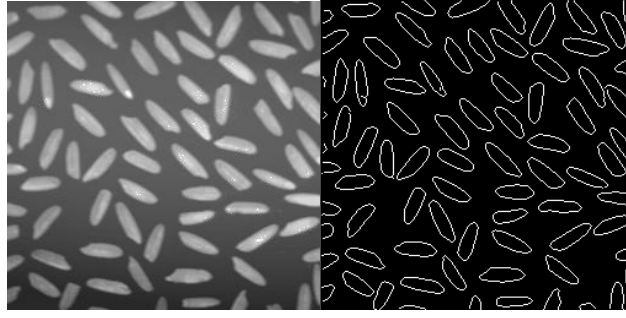


Figura 6. Detección de bordes mediante el filtro de Canny.

4.2 Métodos basados en píxeles

Consideremos la imagen digital $\{f(m,n) : (m,n) \in D \subset \mathbb{Z} \times \mathbb{Z}\}$ donde el dominio $D = \{(m,n) : m=0,1,\dots,M-1; n=0,1,\dots,N-1\}$ se ha dividido en k regiones disjuntas R_1, \dots, R_k , de manera que para un cierto predicado P que asegura la homogeneidad de la región se verifica:

$$\begin{aligned} P(R_i) &= \text{VERDADERO} && \text{para } i=1,2,\dots,k \\ P(R_i \cup R_j) &= \text{FALSO} && \text{para } i \neq j \end{aligned}$$

Esto expresa que la unión de regiones no puede darnos una región homogénea. Dichas regiones deben cubrir el dominio de la imagen, D , es decir, $D = \cup R_i$, y cada píxel sólo puede pertenecer a una región ($R_i \cap R_j = \emptyset$, para $i \neq j$). Al predicado lógico P definido sobre el conjunto de las partes del dominio de la imagen, D , lo llamaremos *regla de segmentación*.

El predicado lógico P suele ser de la forma $P(R, \mathbf{x}, \mathbf{t})$, donde R es una región del dominio, \mathbf{x} es un vector de características de un píxel de la imagen (nivel de gris, $f(m,n)$, textura, etc.) y \mathbf{t} es un conjunto de parámetros (umbrales). Así, una regla de segmentación simple es:

$$P(R, f, t): \quad t_1 < f(m,n) < t_2$$

que expresa que la región R está formada por los píxeles cuyo nivel de gris o intensidad luminosa está en el intervalo (t_1, t_2) .

4.2.1 Segmentación basada en el histograma

El problema más sencillo de segmentación se presenta cuando la imagen está formada por un sólo objeto que tiene intensidad luminosa homogénea sobre un fondo con un nivel de intensidad diferente. En este caso la imagen se puede segmentar en dos regiones utilizando el operador umbral con parámetro t . La elección del valor umbral t se puede hacer a partir del histograma. Si el fondo tiene también intensidad luminosa homogénea, entonces el histograma es bimodal y el umbral t que se debe tomar es el que corresponde al mínimo local que está entre los dos máximos del histograma. En ocasiones no se distinguen bien las dos zonas modales (los máximos) y se debe aplicar una técnica de variación espacial del umbral que consiste en dividir la imagen en bloques cuadrados (ventanas) a los que se les calcula el histograma y el valor

umbral correspondiente (cuando no es bimodal se determina entonces el umbral por interpolación de los valores umbrales de las ventanas de su entorno).

Cuando el histograma es muy irregular, es difícil calcular este mínimo local, en este caso es conveniente suavizarlo (utilizando un filtro de paso-baja).

Si la imagen contiene N objetos (regiones homogéneas) se puede utilizar una umbralización múltiple, es decir,

$$P(R_i) = \text{VERDADERO} \text{ siempre y cuando } R_i = \{(m,n): t_{i-1} \leq f(m,n) \leq t_i\}$$

Los valores umbrales se pueden obtener también del histograma calculando los $N-1$ mínimos locales entre los N picos consecutivos del histograma. A veces estos picos o crestas no se distinguen bien, debido a los píxeles que están en las transiciones de los contornos. Por ello, es conveniente excluir a estos píxeles (bordes locales) en el cálculo del histograma. En la figura 7 se muestra una imagen de la luna. A partir del histograma se observa que los píxeles del intervalo $[75, 255]$ corresponden a la luna y el resto al fondo de la imagen.

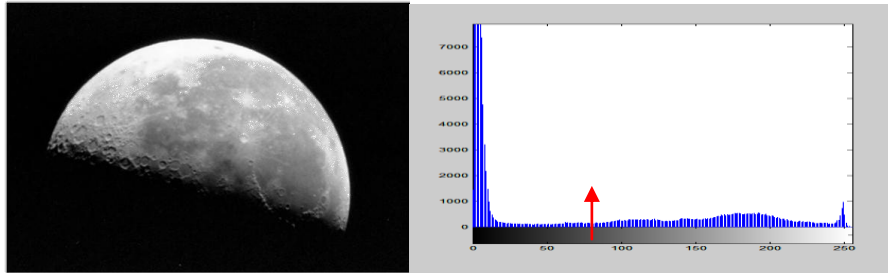


Figura 7. (a) Imagen de la luna. (b) Histograma.

4.2 Técnica basada en las proyecciones

Se basa en las proyecciones de los niveles de gris sobre los ejes de coordenadas, es decir, en la funciones

$$H(m) = \sum_{j=0}^{N-1} f(m, j), \quad m = 0, 1, \dots, M-1$$

$$V(n) = \sum_{i=0}^{M-1} f(i, n), \quad n = 0, 1, \dots, N-1$$

Consiste en determinar un conjunto de subintervalos disjuntos del intervalo $[0, M-1]$, aquellos cuyos puntos tienen mayores valores de $H(m)$. Análogamente se determinan los subintervalos del intervalo $[0, N-1]$ con mayores valores de $V(n)$. Si $[h_1, h_2]$ es un subintervalo horizontal y $[v_1, v_2]$ es un subintervalo vertical entonces puede haber un objeto en el rectángulo $[h_1, h_2] \times [v_1, v_2]$ (ver la figura 8).

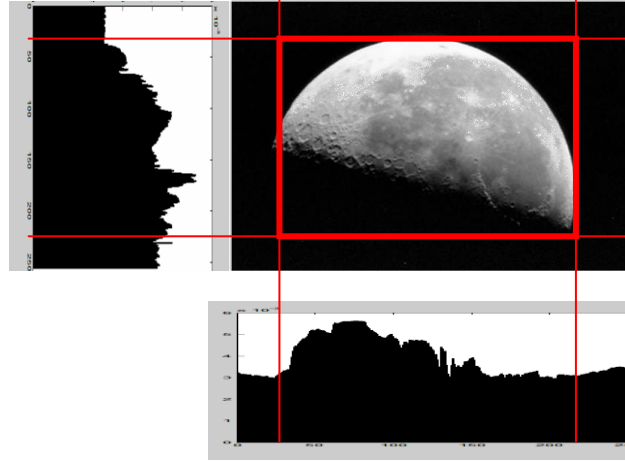


Figura 8. Localización de un objeto basada en sus proyecciones.

4.3 Métodos basados en bordes

En la práctica, el conjunto de píxeles que están en la frontera raramente caracterizan completamente el contorno de un objeto, debido a varias causas, como el ruido o una iluminación no uniforme, que introducen discontinuidades espurias en los tonos de gris. En la segmentación basada en píxeles se suele producir un sesgo en el tamaño de los objetos segmentados, es decir, el tamaño de los objetos va a depender del valor umbral seleccionado, debido a que el tono de gris de los bordes de un objeto puede cambiar gradualmente desde el fondo hasta el valor del objeto. Para evitar esto se pueden utilizar los métodos basados en las s. Estos métodos son secuenciales, van siguiendo el contorno de los objetos utilizando las posiciones de los bordes que vienen determinadas por los valores extremos del gradiente o por los cruces con el cero (si utilizamos operadores de segundo orden). Parten de un píxel del contorno y van determinando la sucesión correspondiente de bordes locales hasta llegar de nuevo al píxel inicial.

Para trazar el contorno de un objeto nos fijamos en el gradiente local de cada píxel que sea un borde local y en su orientación. La cantidad $|\nabla f(i, j) - \nabla f(r, s)|$ se puede utilizar como una medida de discrepancia entre los bordes locales del entorno. También, la cantidad $|\phi(i, j) - \phi(r, s)|$, que es la curvatura local de contorno, se puede utilizar como una medida de discrepancia según las diferentes orientaciones de los bordes locales del entorno, siendo

$$\phi(i, j) = \arctan(D_y^+(i, j) / D_x^+(i, j))$$

la dirección del borde local (i, j) .

A partir de estos valores podemos llevar a cabo el **trazado del contorno**. Así, el píxel (i, j) y su vecino (r, s) , donde $r=i\pm 1, s=j\pm 1$, se pueden enlazar (forman parte de un mismo contorno) si

$$|\nabla f(i, j) - \nabla f(r, s)| \leq T_1 \quad (1)$$

y

$$|\phi(i, j) - \phi(r, s)| \leq T_2 \quad (2)$$

donde T_1 y T_2 son los valores umbrales para el gradiente y el ángulo de orientación, respectivamente. En caso de que hubiera varios píxeles posibles para enlazar se elige el de menor discrepancia (según las dos medidas de discrepancia dadas). Así, un algoritmo para el trazado de contornos puede ser el siguiente:

ALGORITMO: Trazado de contornos

1. Supongamos que el contorno se ha determinado hasta el elemento (i, j) .
2. Se elige el píxel (r, s) adyacente a (i, j) en la dirección $\phi(i, j)$. Si (r, s) verifica la ecuación (1) entonces se considera un elemento del contorno y se regresa al paso 1. Si no, se va a paso 3.
3. Determinar el nivel medio de gris en el entorno 3×3 del elemento (r, s) y compararlo con algún valor preestablecido para saber si (r, s) está dentro o fuera de la región. Ir al paso 4.
4. Seleccionar el píxel (h, k) adyacente a (i, j) en la dirección $\phi(i, j) \pm \pi/4$ (el signo se determina según el resultado del paso 3). Si verifica (1) entonces se considera un elemento del borde, continuación del anterior. Si no, comenzar el trazado del contorno con otro píxel más prometedor.

4.4 Métodos de segmentación basados en crecimiento de regiones.

Las técnicas basadas en píxeles aseguran la homogeneidad de las regiones pero no garantizan que sean conexas. En la segmentación de imágenes se debe considerar también la información suministrada por los píxeles del entorno para conseguir así regiones homogéneas y conexas, pues los píxeles de un mismo entorno suelen tener propiedades estadísticas similares y pertenecer a una misma región.

Una segmentación basada en regiones de una imagen digital I consiste en realizar una partición de la imagen en k regiones con las siguientes propiedades:

- a) Las regiones obtenidas en la partición, R_1, R_2, \dots, R_k , deben ser disjuntas

$$R_i \cap R_j = \phi, \quad i \neq j.$$

- b) Su unión debe ser la imagen completa, $\bigcup_{i=1}^k R_i = I$.
- c) Cada región R_i tiene que ser conexa, es decir, todos sus píxeles conectados
- d) Se debe verificar que:

$$P(R_i) = \text{VERDADERO} \quad \text{y} \quad P(R_i \cup R_j) = \text{FALSO},$$

para regiones adyacentes cualesquiera, R_i y R_j , siendo P el predicado que nos proporciona el **test de homogeneidad** de la región.

La condición a) nos indica que cada píxel sólo puede pertenecer a una región; la condición c) que cada región debe estar formada por píxeles conectados según algún criterio predefinido, como por ejemplo, que no puede haber píxeles cuyo entorno 3×3 no contenga ningún otro píxel de la región; la condición d) establece la propiedad (o propiedades) que tienen que cumplir todos los píxeles de esa región y que además las regiones adyacentes tienen que ser diferentes según dicha propiedad.

El **crecimiento de las regiones** es un procedimiento que consiste en ir formando grupos (regiones) de píxeles por incorporación sucesiva de píxeles de la imagen a los grupos según algún criterio predefinido. El procedimiento más sencillo de **crecimiento de regiones** comienza con un conjunto de píxeles, llamados *semillas*, que representan las distintas regiones de la imagen (al menos una semilla por región). A partir de las semillas se van incorporando nuevos píxeles a las regiones utilizando un *mecanismo de crecimiento* que detecta, en cada etapa k y para cada región $R_i(k)$, los píxeles aun no clasificados que pertenecen a un entorno predefinido de algún píxel del contorno de la región $R_i(k)$. Para cada píxel (x,y) , detectado de esta forma, se comprueba si cumple la *regla de homogeneidad*, es decir, si la nueva región $D_i(k) \cup (x,y)$ sigue siendo homogénea, en cuyo caso se consigue ampliar la región añadiéndole dicho píxel. El píxel se elige del entorno de un píxel del contorno de la región para garantizar de alguna manera la conectividad de la región.

Cuando las imágenes son monocromáticas el análisis de la región se hace utilizando un conjunto de descriptores como el tono de gris medio, textura, momentos, propiedades espaciales, etc. La regla de homogeneidad obedecerá a algún criterio de similitud de los píxeles de la misma región y se basa en dichos descriptores.

La solución dada por esta técnica depende en gran medida de las semillas elegidas. Por ello es conveniente elegir bien dichas semillas. Los píxeles que corresponde a los picos del histograma pueden ser una buena elección.

Otra cuestión importante es la formulación de una regla de parada. El procedimiento de parada termina cuando no quedan más píxeles para que sean asignados a una región pues los criterios de similitud suelen ser de naturaleza local y no se modifican durante el proceso de crecimiento, es decir, no tiene en cuenta la historia del proceso. También se pueden utilizar criterios de similitud dinámicos que tienen en cuenta conceptos como el tamaño o la forma de la región y la representatividad de un píxel con respecto a los píxeles de la región.

Cuando se elige más de una semilla por región es necesario unir o fusionar algunas de las regiones obtenidas. Está claro que las regiones a unir deben de ser adyacentes y tener propiedades estadísticas similares, como pueden ser la intensidad (nivel de gris) media de la región y su desviación típica. Por ejemplo, una regla de unión pueden ser que los valores medios y varianzas de las regiones i y j verifique:

$$|m_i - m_j| < k\sigma_i \quad y \quad |m_i - m_j| < c\sigma_j$$

donde m_i y σ_i son la media y la desviación típica de la región R_i , respectivamente, y c una constante predefinida, como puede ser $c = 3$.

La fusión de regiones se puede incorporar en el mecanismo de crecimiento y conseguir así un algoritmo combinado de crecimiento y fusión. Para ello se necesita de una regla de fusión, de la forma $P(R_i \cup (r,s))$, que especifica cuando un píxel (r,s) se asigna a una de sus regiones R_i adyacentes:

$$P(R_i \cup (r,s)) = \begin{cases} \text{VERDADERO} & \text{si } |f(r,s) - m_i| \leq T_i(r,s) \\ \text{FALSO} & \text{en otro caso.} \end{cases}$$

donde m_i es el valor medio de la intensidad luminosa de la Región R_i , y $T_i(r,s)$ es el valor umbral utilizado para dicha región R_i . Se puede tomar,

$$T_i(r,s) = \left(1 - \frac{\sigma_i^*}{m_i^*}\right)T$$

donde m_i^* y σ_i^* son la media y la desviación típica, respectivamente, de los valores de intensidad de los píxeles de $R_i \cup (r,s)$. Si más de una región R_i cumple la regla entonces se asigna a la región con valor $|f(r,s) - m_i|$ mínimo. Si no existen regiones adyacentes o no cumple la regla de fusión entonces el píxel (r,s) se asigna a una nueva región.

El crecimiento de la región depende en gran medida del valor umbral T seleccionado. Si T es pequeño es más difícil que se produzcan fusiones y se generan muchas regiones pequeñas, en cambio, si T es grande se favorece la fusión y así se generan pocas regiones pero grandes (con menos homogeneidad y mayor desviación típica). Asimismo, como $T_i(r,s)$ depende del cociente σ_i^*/m_i^* , cuando las regiones son homogéneas σ_i^*/m_i^* tiende a cero y T_i tiende a T . Por tanto, T es la diferencia máxima admisible entre $f(r,s)$ y m_i para fusionar el píxel (r,s) con una región bastante homogénea.

4.5 División y fusión de regiones

Los procedimientos de crecimiento de regiones parten de un conjunto de píxeles semilla. Un método alternativo se obtiene actuando al revés, es decir, partiendo de la imagen completa se van realizando sucesivas divisiones de la misma. Cada subimagen obtenida que no sea homogénea se vuelve a dividir y así sucesivamente. Así, por ejemplo, se puede comenzar con una imagen inicial. Esta se divide en cuatro subimágenes, llamadas cuadrantes, y cada subimagen obtenida que no sea homogénea se vuelve a dividir en otras cuatro subimágenes; el proceso se continúa sucesivamente hasta que todas ellas sean homogéneas, es decir, cumplan las condiciones (a), (b), (c) y (d) establecidas para que constituyan una segmentación. Sólo se necesita aplicar la regla de homogeneidad en cada cuadrante. Si el predicado P es FALSO para algún cuadrante entonces se subdivide en 4 subcuadrantes y así sucesivamente. Dichos métodos se llaman de partición o división. Si la imagen tiene una dimensión $N \times N$ y N es potencia de 2, es decir, $N=2^n$, el procedimiento recursivo se puede representar mediante un *árbol cuaternario* (cada vértice tiene 4 descendientes) como se muestra en la figura 9, donde el nodo raíz corresponde a la imagen completa y sus descendientes son cada uno de los 4 cuadrantes en que se divide, y así sucesivamente.

Las técnicas basadas en división tienen el inconveniente de que pueden generar regiones que sean adyacentes y con idénticas propiedades (homogéneas), que deberían estar fundidas en una sólo región. Este inconveniente sugiere que se utilicen técnicas mixtas, es decir, algoritmos de partición (división) y fusión.

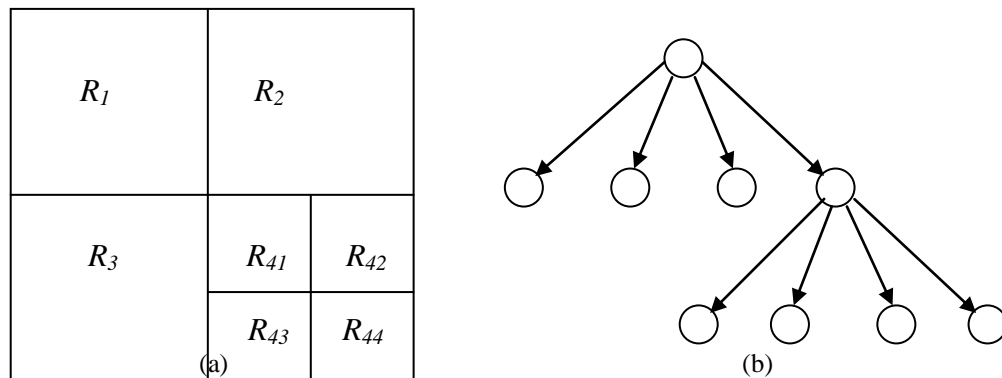


Figura 9. (a) Imagen dividida. (b) Árbol cuaternario correspondiente.

Un **algoritmo de partición y fusión** consta de los siguientes pasos:

1. Dividir en cuatro cuadrantes disjuntos cada región R_i con $P(R_i) = \text{FALSO}$.
- 2.- Cuando no sean posible más divisiones fundir las regiones adyacentes R_i y R_j para las cuales $P(R_i \cup R_j) = \text{VERDADERO}$.
- 3.- Parar cuando no se puedan hacer más fusiones.

4.6 Métodos de segmentación basados en modelos: la transformada de Hough

La mayoría de las técnicas anteriores utilizan sólo información local. La información global se puede introducir a través de modelos. Con este método se trata de enlazar los bordes locales cuando ellos están sobre una curva bien especificada (rectas, rectángulos, circunferencias, elipses u otro tipo de curvas en una imagen que se puedan expresar por un número reducido de parámetros). Para ello vamos a utilizar técnicas basadas en la transformación de Hough, que se puede utilizar con éxito en segmentación de objetos solapados o parcialmente ocluidos.

a) Detección de rectas

Consideremos una imagen $f(x,y)$ de tamaño $M \times N$ y sea $e(x,y)$ la imagen que se obtiene después de aplicarle un filtro de paso-alta, el operador valor absoluto y el operador umbral para obtener los píxeles que son bordes locales. Supongamos que la imagen $e(x,y)$ tiene n píxeles en tono blanco que identifican los bordes locales. Para encontrar todos los segmentos (trozos de una recta que puede corresponder a bordes de un objeto parcialmente ocluido) que aparecen en la imagen gradiente podemos determinar las $n(n-1)/2$ rectas (a lo sumo) que pasan por dos o más puntos y comprobar cuántos puntos verifican la ecuación de cada recta. El número de operaciones elementales que se

requiere para ello es de orden n^3 . Cuando n es grande, el proceso computacional puede ser prohibitivo. Hough (1962) propuso el siguiente método para determinar los trozos de rectas en una imagen.

La ecuación normal de una recta viene dada por la expresión:

$$x \cos \theta + y \sin \theta = r \quad (3)$$

donde θ es el ángulo que forma la recta perpendicular a ella que pasa por el origen con el eje de abscisas ($\theta \in [-\pi, \pi]$) y r ($r \geq 0$) es la distancia de la recta al origen de coordenadas (ver figura 10). Se utiliza dicha ecuación porque contempla también a las rectas verticales (la ecuación del tipo $y=a+bx$ no contempla rectas con pendiente infinita, por ejemplo la recta $x=c$).

Un píxel con coordenadas (x_i, y_i) está sobre la recta (3) si verifica

$$x_i \cos \theta + y_i \sin \theta = r \quad (4)$$

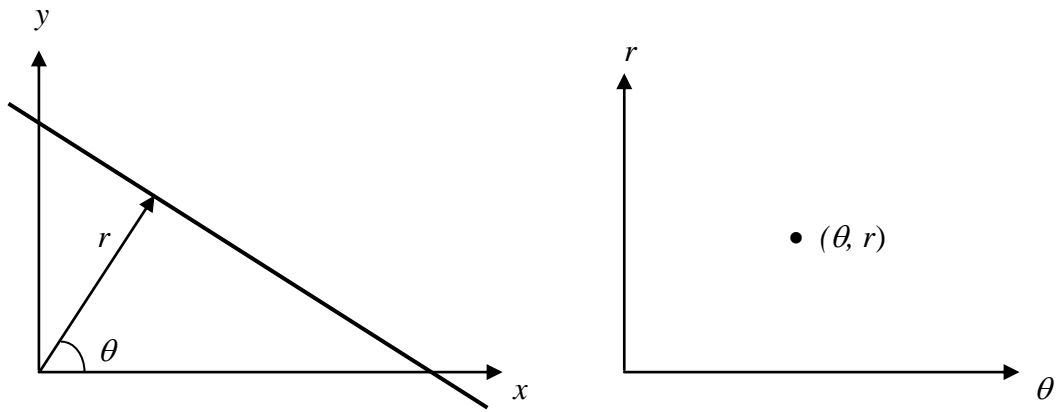


Figura 10. Representación polar de una recta

Si ahora consideramos un nuevo espacio donde las variables son θ y r , la ecuación (4) nos da la relación que existe entre los valores de θ y r de todas las rectas que pasan por el punto (x_i, y_i) y cuya representación gráfica corresponde a una curva sinusoidal. En la figura 11 mostramos la curva que corresponde a todos los pares (θ, r) que verifican la ecuación (4) cuando $(x_i, y_i) = (30, 20)$, donde sólo los puntos de la curva con $r \geq 0$ corresponden a rectas que pasan por dicho punto. Es decir, el conjunto

$$\{ (\theta, r) : 30 \cos \theta + 20 \sin \theta = r, \theta \in [-\pi, \pi] \text{ y } r \geq 0 \}$$

nos determina el haz de todas las rectas que pasan por el punto $(30, 20)$.

La transformación *punto a curva* que aplica cada punto del plano cartesiano (X,Y) en una curva sinusoidal del plano (θ, r) , donde los puntos de dicha curva se corresponde con las rectas del haz de todas las rectas que pasan por el punto del plano cartesiano, se conoce con el nombre de la **transformación de Hough**.

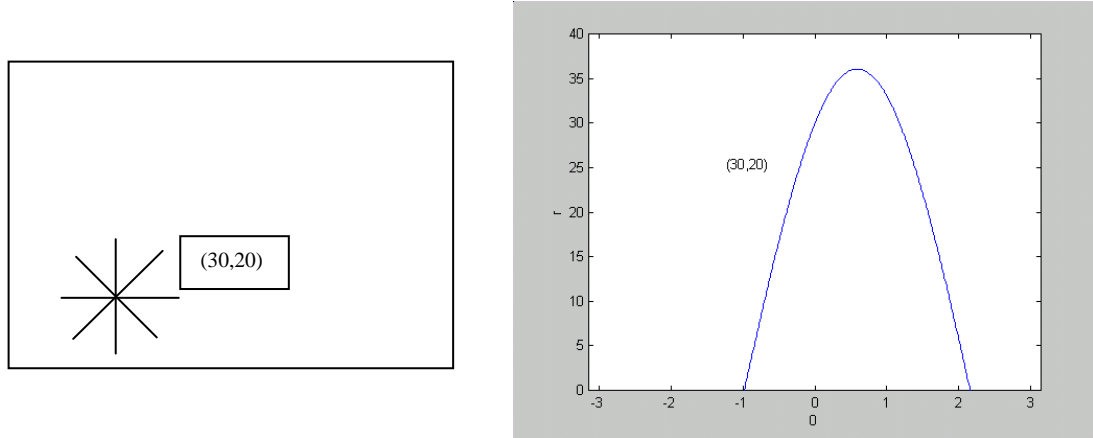


Figura 11. Representación gráfica de la relación (4) correspondiente al punto (30,20).

Puede observarse que si tenemos n puntos alineados en el plano cartesiano, sus curvas correspondientes en el plano (r, θ) se cortan en un mismo punto (θ_n, r_n) , que nos da los valores de los parámetros de la ecuación de la recta que pasa por los n puntos. Por lo tanto, cada recta del plano cartesiano se aplica también en un punto del plano (θ, r) .

Como las posiciones de los píxeles de la imagen podemos suponer que están en el primer cuadrante, las rectas que tiene un valor de $\theta \in (-\pi/2, -\pi)$ no pueden pasar por píxeles de la imagen. Por ello, sólo estaremos interesados en los valores de $\theta \in [-\pi/2, \pi]$. La figura 12 muestra las curvas sinusoidales correspondientes a los píxeles (200,400), (400,200) y (600,0) de una imagen 640×480. Como están alineados, dichas curvas se cortan en un mismo punto.

Para detectar grupos de puntos alineados construimos las curvas correspondientes en el espacio (r, θ) de cada uno de los puntos que tenemos en el espacio (x,y) y almacenamos los puntos del espacio (r, θ) que son intersección de t o más curvas, donde t es un valor umbral apropiado.

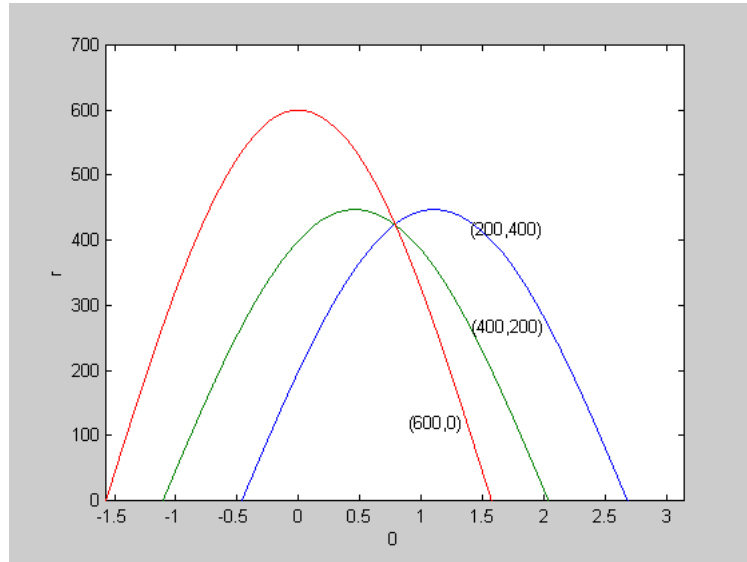


Figura 12. Transformadas de Hough de los puntos (200,400), (400,200) y (600,0).

Para implementar esta técnica en un ordenador, con una imagen de tamaño $M \times N$, los puntos (píxeles) pertenecerán al conjunto

$$\{ (m,n) : m=0,1,2,\dots,M-1, n=0,1,2,\dots,N-1 \}$$

y no puede haber rectas que pasen por alguno de dichos puntos que tengan un valor de θ dentro del intervalo $(-\pi/2, -\pi)$, ya que estos puntos están todos en el primer cuadrante. Asimismo, el valor máximo de r que puede alcanzar una de estas rectas es $(M^2 + N^2)^{1/2}$. Por lo tanto, el parámetro r va a tomar los valores $0,1,\dots,(M^2 + N^2)^{1/2}$ y el parámetro θ un conjunto de valores uniformemente repartidos en el intervalo $[-\pi/2, \pi]$, cuyo número depende de la precisión que se desee obtener. Para cada valor r_j del parámetro r y θ_k del parámetro θ , se define un *acumulador* $A(j, k)$. Se elige un punto (x_i, y_i) y un valor θ_k , y se determina el valor correspondiente de r en la ecuación (4). Si dicho valor es r_j (positivo) entonces se incrementa en una unidad el acumulador correspondiente,

$$A(i, j) \leftarrow A(i, j) + 1$$

Los acumuladores que corresponden a máximos locales (con respecto a los acumuladores de su entorno) y que superan el valor umbral T nos determinan los parámetros de las rectas detectadas.

Una propiedad importante de la transformada de Hough es su insensibilidad a los trozos que faltan de la recta, al ruido de la imagen y a otras estructuras no lineales coexistentes en la imagen. Los puntos que no estén bien alineados no se transformarán

en un punto en el espacio paramétrico, sino que formarán grupos de puntos; su centro de gravedad se puede considerar la representación de la recta.

En la figura 13 se muestran las transformadas de Hough de 40 puntos alineados pertenecientes a la ecuación de la recta $y = -x + 80$ cuya ecuación normal es $x\cos(\pi/4) + y\sin(\pi/4) = 56,5685$. Puede observarse como todas las curvas sinusoidales se cortan en el punto $(\pi/4, 56,5685)$ del espacio paramétrico.

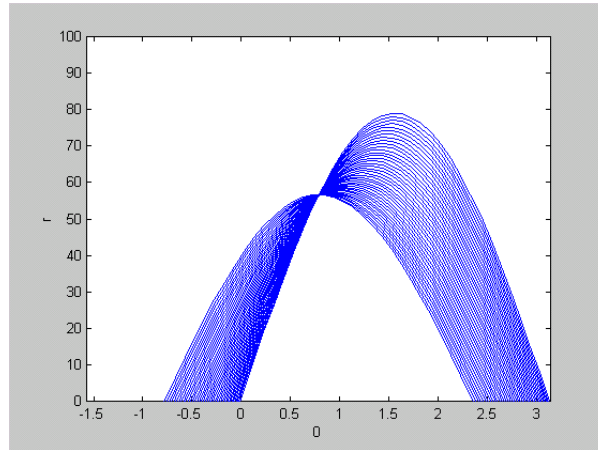


Figura 13. Transformadas de Hough de 40 puntos alineados.

b) Detección de circunferencias

La ecuación de una circunferencia de radio r , centrada en el punto (a,b) que pasa por el punto (x_i,y_i) viene dada por la expresión:

$$(x_i - a)^2 + (y_i - b)^2 = r^2 \quad (5)$$

Los parámetros (a,b,r) nos permiten describir todas las circunferencias que pasan por (x_i,y_i) . Así, la transformada de Hough aplica cada punto de R^2 en el conjunto de puntos del espacio paramétrico (a,b,r) que verifican la ecuación (5), es decir, una familia de circunferencias con centro en (x_i,y_i) para cada valor de r . Si deseamos determinar todas las circunferencias que aparecen en la imagen con un radio dado, r_o , entonces aplicamos cada punto (x_i,y_i) en la familia de circunferencias de radio r_o y centro en dicho punto.

Si los puntos del conjunto $\{(x_j,y_j), j=1,2,...,n\}$ pertenecen a una misma circunferencia con centro (a_o,b_o) y radio r_o entonces sus correspondientes circunferencias, según la transformación de Hough, se cortan en el punto (a_o,b_o) (ver figura 14).

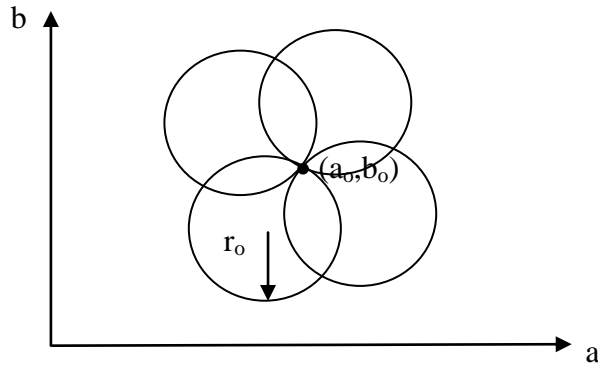


Figura 14. Transformadas de Hough de 4 puntos que pertenecen a una circunferencia de centro (a_o, b_o) y radio r_o .

Para determinar todas las circunferencias (o tramos de circunferencia) que aparecen en una imagen determinamos en primer lugar los valores posibles de los parámetros a , b y r (cuantificación del espacio paramétrico) es decir, $a \in \{a_1, \dots, a_p\}$, $b \in \{b_1, \dots, b_q\}$ y $r \in \{r_1, \dots, r_h\}$. Definimos los *acumuladores*

$$A(a, b, r), \quad \forall a \in \{a_1, \dots, a_p\}, b \in \{b_1, \dots, b_q\} \text{ y } r \in \{r_1, \dots, r_h\}.$$

Inicialmente con valor cero. Para cada borde local (x_i, y_i) y para cada valor de a y b determinamos el valor de r a partir de la ecuación (5) e incrementamos al acumulador correspondiente en una unidad. Una vez recorridos todos los puntos y los diferentes valores de a y b , determinamos los máximos locales en la matriz de acumuladores, es decir, seleccionamos aquellas celdas cuyo acumulador tiene el mayor valor de su entorno y después nos quedamos sólo con aquellos acumuladores con valor superior a un cierto valor umbral T . Estos acumuladores nos determinan los parámetros de las circunferencias que aparecen en la imagen.

En general, para una curva de ecuación $f(x, y, \theta_1, \dots, \theta_p) = 0$, con p parámetros, tendríamos el siguiente algoritmo:

ALGORITMO: Detección de curvas usando la transformada de Hough

1. **Cuantificar** el espacio de los parámetros.
2. **Inicializar** a cero todos los acumuladores $A(\theta_1, \dots, \theta_p)$ posibles obtenidos para los diferentes valores del vector paramétrico $(\theta_1, \dots, \theta_p)$ (según los valores cuantificados de los parámetros).
3. Para cada píxel (x_i, y_i) de la imagen de bordes locales y para los diferentes valores posibles del vector paramétrico **incrementar el acumulador** $A(\theta_1, \dots, \theta_p)$ en una unidad,

$$A(\theta_1, \theta_2, \dots, \theta_p) \leftarrow A(\theta_1, \theta_2, \dots, \theta_p) + 1$$
 siempre y cuando se verifique la ecuación $f(x_i, y_i, \theta_1, \dots, \theta_p) = 0$.

4. **Determinar los máximos locales** en la tabla de acumuladores y aplicarles el operador umbral. Los acumuladores que superen el umbral nos dan los parámetros de las curvas detectadas que verifican la ecuación $f(x,y, \theta_1, \dots, \theta_p)=0$, que están presentes en la imagen original.

Finalmente, como este método requiere una gran cantidad de cómputo, podemos introducir información adicional acerca de las direcciones de los bordes para conseguir una reducción sustancial de la misma. Así, con el conocimiento de la dirección, sólo un número reducido de acumuladores se irán incrementando. Por ejemplo, si deseamos detectar rectas y utilizamos entornos de tamaño 9 entonces podemos asignar una de las 8 direcciones posibles a cada borde. De esta manera sólo se incrementará el acumulador si el borde local tiene la dirección apropiada y sólo habría que recorrer los bordes locales con la dirección apropiada. Si queremos, además, que nuestro procedimiento sea poco sensible al ruido, podemos admitir direcciones que varíen en $\pm 45^\circ$.

Si deseamos detectar el contorno de un círculo negro de radio R , con fondo blanco, entonces todos los acumuladores, $A(a,b)$, cuyo punto (a,b) está en el círculo de centro (x_i, y_i) y radio R incrementarían su valor. En cambio, si utilizamos la información de la dirección de los bordes locales, de manera que $\phi(x,y)$ nos da la dirección del borde local (x,y) , entonces sólo se incrementaría aquellos acumuladores cuyos parámetros a y b se pueden identificar a partir de la siguiente formula:

$$a = x_i - R \cos(\phi(x,y))$$

$$b = y_i - R \sin(\phi(x,y))$$

$$\phi(x,y) \in [\phi(x,y) - \pi/4, \phi(x,y) + \pi/4]$$

Se establece que para que un borde local (x_i, y_i) forme parte del círculo de radio R y centro (a,b) debe tener un dirección $\phi(x,y)$ igual a la dirección del vector $(x_i - a, y_i - b)$, admitiéndose de antemano un error máximo en torno a dicha dirección.