

Técnicas Clásicas de Segmentación de Imagen

Marcos Martín

17 de marzo de 2010

1. Introducción

El primer paso en cualquier proceso de análisis de imagen es la segmentación. Mediante la segmentación vamos a dividir la imagen en las partes u objetos que la forman. El nivel al que se realiza esta subdivisión depende de la aplicación en particular, es decir, la segmentación terminará cuando se hayan detectado todos los objetos de interés para la aplicación. En general, la segmentación automática es una de las tareas más complicadas dentro del procesado de imagen. La segmentación va a dar lugar en última instancia al éxito o fallo del proceso de análisis. En la mayor parte de los casos, una buena segmentación dará lugar a una solución correcta, por lo que, se debe poner todo el esfuerzo posible en la etapa de segmentación.

Los algoritmos de segmentación de imagen generalmente se basan en dos propiedades básicas de los niveles de gris de la imagen: discontinuidad y similitud. Dentro de la primera categoría se intenta dividir la imagen basándonos en los cambios bruscos en el nivel de gris. Las áreas de interés en esta categoría son la detección de puntos, de líneas y de bordes en la imagen. Las áreas dentro de la segunda categoría están basadas en las técnicas de umbrales, crecimiento de regiones, y técnicas de división y fusión.

2. Detección de Discontinuidades

2.1. Generalidades

En esta sección vamos a presentar varias técnicas para detectar varios tipos de discontinuidades: puntos, líneas y bordes. El método más común de buscar discontinuidades es la correlación de la imagen con una máscara. En la figura 1 se puede ver un caso general de máscara de 3×3 . En este procedimiento se realiza el producto de los elementos de la máscara por el valor de gris correspondiente a los pixels de la imagen encerrados por la máscara. La respuesta a la máscara de cualquier pixel de la imagen viene dado por

$$R = \sum_{i=1}^9 w_i z_i \quad (1)$$

donde z_i es el nivel de gris asociado al pixel de la imagen con coeficiente de la máscara w_i . Como suele ser habitual, la respuesta de la máscara viene referida a

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 1: Máscara 3×3 .

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

Figura 2: Máscara usada para la detección de puntos aislados.

su posición central. Cuando la máscara esté centrada en un pixel de borde de la imagen, la respuesta se determina empleando el vecindario parcial apropiado.

2.2. Detección de Puntos

La detección de puntos aislados es inmediata. Empleando la máscara de la figura 2, vamos a decir que se ha detectado un punto en la posición en la cual está centrada la máscara si

$$|R| > T \quad (2)$$

donde T es un umbral. Básicamente se mide la diferencia entre el pixel central y sus vecinos, puesto que un pixel será un punto aislado siempre que sea suficientemente distinto de sus vecinos. Solamente se considerarán puntos aislados aquellos cuya diferencia con respecto a sus vecinos sea significativa.

2.3. Detección de Líneas

En este caso se consideran las máscaras de la figura 3. Si pasamos la primera de las máscaras a lo largo de la imagen, tendrá mayor respuesta para líneas de ancho un pixel orientadas horizontalmente. Siempre que el fondo sea uniforme, la respuesta será máxima cuando la línea pase a lo largo de la segunda fila de la máscara. La segunda máscara de la figura 3 responderá mejor a líneas

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

Horizontal

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

45°

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

Vertical

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

-45°

Figura 3: Máscaras de línea.

orientadas a 45°; la tercera máscara a líneas verticales; y la última a líneas orientadas a -45°. Estas direcciones se puede establecer observando que para la dirección de interés las máscaras presentan valores mayores que para otras posibles direcciones. Si denotamos con R_1 , R_2 , R_3 y R_4 las respuestas de las cuatro máscaras de la figura 3 para un pixel en particular, entonces si se cumple que $|R_i| > |R_j|$ con $j \neq i$, será más probable que dicho pixel esté asociado a la dirección correspondiente a la máscara i .

2.4. Detección de Bordes

2.4.1. Preliminares

La detección de bordes es el procedimiento empleado más habitualmente para la detección de discontinuidades. Un borde se define como la frontera entre dos regiones con nivel de gris relativamente diferente. Vamos a suponer a partir de ahora que las regiones de interés son suficientemente homogéneas de modo que la transición entre dichas regiones se puede determinar empleando exclusivamente las discontinuidades en el nivel de gris.

La idea básica detrás de cualquier detector de bordes es el cálculo de un operador local de derivación. En la figura 4 se puede ver este concepto. En la parte derecha se puede ver una imagen de una banda clara sobre un fondo oscuro, el perfil a lo largo de una línea horizontal y la primera y segunda derivada de dicho perfil. Se puede observar que el perfil del borde se ha modelado como una discontinuidad suave. Esto tiene en cuenta el hecho de que en las imágenes reales los bordes están ligeramente desenfocados.

Como se puede observar en la figura 4 la primera derivada es positiva para cambio a nivel de gris más claro, negativa en caso contrario y cero en aquellas zonas con nivel de gris uniforme. La segunda derivada presenta valor positivo en la zona oscura de cada borde, valor negativo en la zona clara de cada borde y valor cero en las zonas de valor de gris constante y justo en la posición de los bordes. El valor de la magnitud de la primera derivada nos sirve para detectar la presencia de bordes, mientras que el signo de la segunda derivada nos indica si el pixel pertenece a la zona clara o a la zona oscura. Además la segunda derivada presenta siempre un cruce por cero en el punto medio de la transición. Esto puede ser muy útil para localizar bordes en una imagen.

Aunque lo que llevamos dicho se refiere a perfiles unidimensionales, la ex-

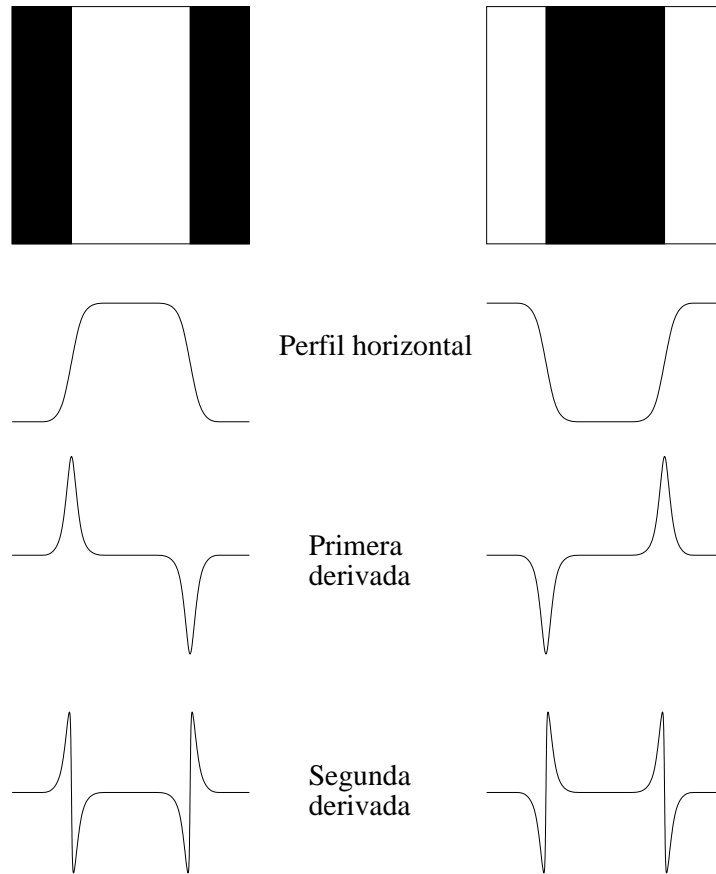


Figura 4: Detección de bordes empleando operadores de derivación. La segunda derivada tiene un cruce por cero en la posición de cada borde.

tensión a dos dimensiones es inmediata. Simplemente se define el perfil en la dirección perpendicular a la dirección del borde y la interpretación anterior seguirá siendo válida. La primera derivada en cualquier punto de la imagen vendrá dada por la magnitud del gradiente, mientras que la segunda derivada vendrá dada por el operador Laplaciano.

2.4.2. Gradiente

El gradiente de una imagen $I(x, y)$ en la posición (x, y) viene dado por el vector

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}. \quad (3)$$

El vector gradiente siempre apunta en la dirección de la máxima variación de la imagen I en el punto (x, y) . En la detección de bordes es muy importante la magnitud de este vector, denominado simplemente como gradiente de la imagen, denotado por ∇I y dado por

$$\nabla I = \|\nabla I\| = \sqrt{I_x^2 + I_y^2}. \quad (4)$$

0	1
-1	0

1	0
0	-1

(a)

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

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

(b)

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

(c)

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

(d)

Figura 5: Operadores de derivación: (a) de Roberts, (b) de Prewitt, (c) de Sobel y (d) de Frei-Chen.

Esta cantidad representa la variación de la imagen $I(x, y)$ por unidad de distancia en la dirección del vector ∇I . En general, el gradiente se suele aproximar mediante la expresión

$$\nabla I \approx |I_x| + |I_y|, \quad (5)$$

que es mucho más simple de implementar en la práctica. La dirección del vector gradiente también es una cantidad importante. Sea $\alpha(x, y)$ el ángulo del vector ∇I en el punto (x, y) . Entonces se tiene que

$$\alpha(x, y) = \tan^{-1} \frac{I_y(x, y)}{I_x(x, y)} \quad (6)$$

donde los ángulos se miden con respecto al eje de abscisas.

El cálculo del gradiente se basa en obtener las derivadas parciales para cada pixel. Las derivadas se pueden implementar digitalmente de varias formas. En

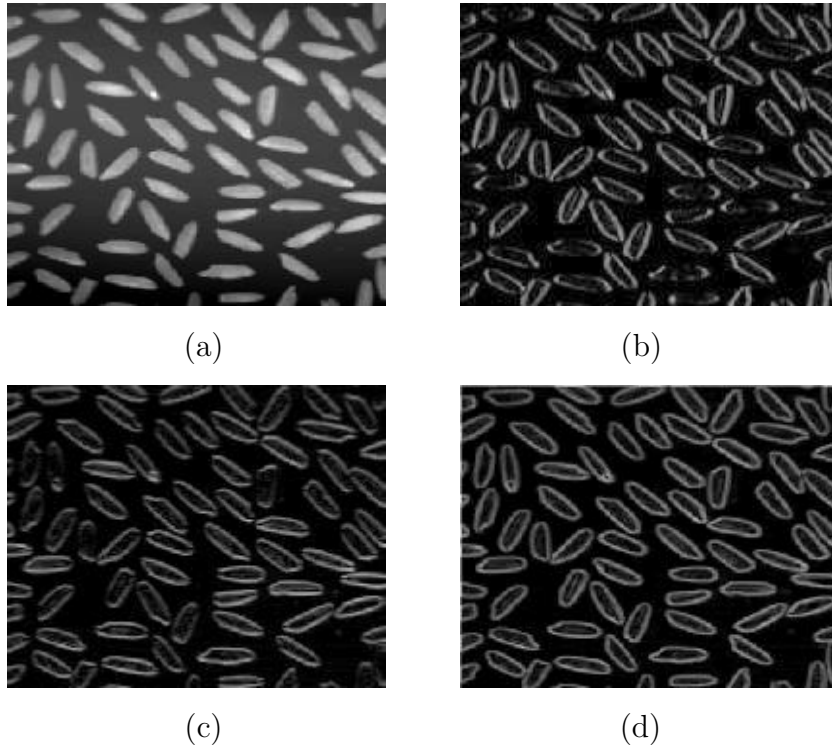


Figura 6: (a) Imagen original. (b) Derivación horizontal usando Sobel. (c) Derivación vertical usando Sobel. (d) Imagen gradiente usando Sobel.

la figura 5 se pueden ver los operadores de Roberts, Prewitt, Sobel y Frei-Chen para determinar las derivadas parciales. Sin embargo, los operadores de Sobel y de Frei-Chen tienen la ventaja de que proporcionan un suavizado además del efecto de derivación. Ya que la derivación acentúa el ruido, el efecto de suavizado es particularmente interesante, puesto que elimina parte del ruido. El requisito básico de un operador de derivación es que la suma de los coeficientes de la máscara sea nula, para que la derivada de una zona uniforme de la imagen sea cero.

Según la figura 5 las derivadas según el operador de Sobel vienen dadas por

$$R_x = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (7)$$

$$R_y = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (8)$$

En la figura 6 podemos ver el resultado de aplicar las máscaras de Sobel a una imagen que contiene granos de arroz.

2.4.3. Laplaciano

El Laplaciano de una imagen $I(x, y)$ es una derivada de orden dos definida por

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}. \quad (9)$$

En general se suele tomar el valor negativo del Laplaciano. Al igual que en el caso del gradiente se puede implementar en forma digital de varias formas.

0	-1	0
-1	4	-1
0	-1	0

Figura 7: Máscara utilizada para calcular el Laplaciano.

Puesto que el Laplaciano es un operador de derivación la suma de los coeficientes debe ser cero. Además, el coeficiente asociado con el pixel central debe ser positivo y los demás coeficientes negativos o ceros. En la figura 7 podemos ver una máscara para el Laplaciano. En este caso la expresión para determinar el Laplaciano viene dada por

$$4z_5 - (z_2 + z_4 + z_6 + z_8). \quad (10)$$

Aunque el Laplaciano responde a transiciones en la intensidad de la imagen, se emplea en pocas ocasiones en la práctica. Debido a que es un operador de segunda derivada es sensible en exceso a la presencia de ruido. Además, el Laplaciano da lugar a bordes dobles, como se puede apreciar en la figura 4 y no permite determinar direcciones. En general juega un papel secundario en la detección de bordes para determinar si un pixel está en la zona clara o en la zona oscura del borde a través del signo del Laplaciano.

Un uso más generalizado del Laplaciano es a través de la propiedad de localización de los bordes usando la propiedad de los cruces por cero (ver figura 4). Este concepto se basa en correlar la imagen con el Laplaciano de una función Gaussiana de la forma

$$h_\sigma(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (11)$$

donde σ es la desviación estándar. Si $r^2 = x^2 + y^2$, el Laplaciano de una función $h(r)$ se puede expresar como

$$\nabla^2 h = \frac{\partial^2 h}{\partial r^2} + \frac{1}{r} \frac{\partial h}{\partial r}, \quad (12)$$

entonces el Laplaciano de $h_\sigma(r)$ (h_σ no depende de la variable angular) se puede poner como

$$\nabla^2 h_\sigma = \left(\frac{r^2 - 2\sigma^2}{\sigma^4}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (13)$$

El negativo de esta función es una función suave que presenta cruces por cero para $\pm\sigma$, valor positivo en el origen y negativo a partir de $\pm\sigma$, como puede verse

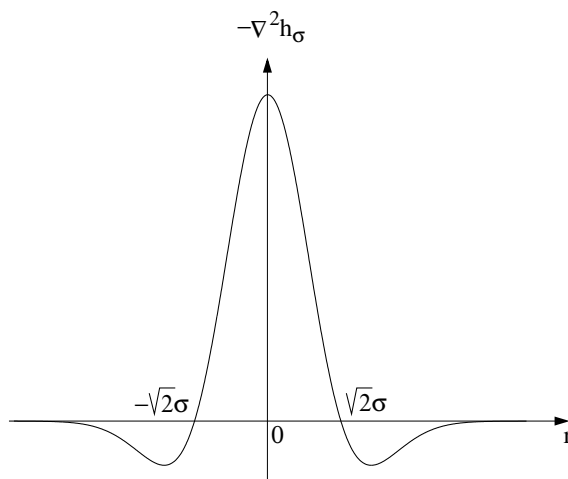


Figura 8: Corte por el origen de $-\nabla^2 h_\sigma$.

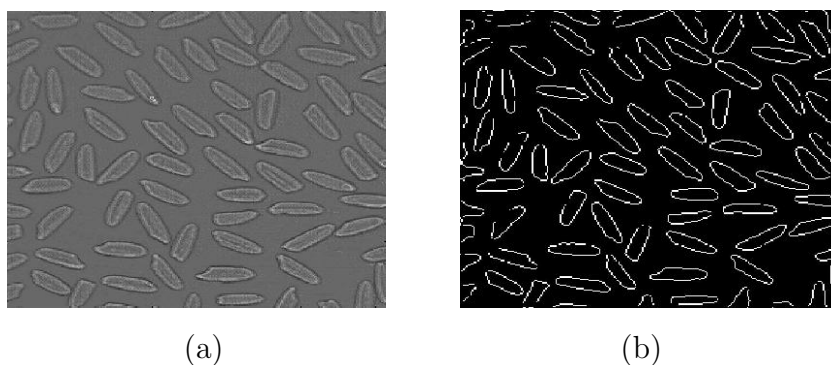


Figura 9: (a) Resultado tras el operador $-\nabla^2 h_\sigma$. (b) Cruces por cero.

en la figura 8. Se puede comprobar que el valor medio del operador $-\nabla^2 h_\sigma$ es cero, al igual que el valor medio de la imagen que se obtiene al emplear este operador. Correlando una imagen con este operador, se suaviza o desenfoca por un factor proporcional al valor de σ empleado. Aunque éste da lugar a una reducción del ruido, su utilidad fundamental es debido a los cruces por cero.

En la figura 9(a) se puede ver el resultado de aplicar el operador $-\nabla^2 h_\sigma$ a la imagen de la figura 6(a). En la figura 9(b) se pueden observar los cruces por cero.

3. Enlazado de Bordes.

3.1. Necesidad

Las técnicas de detección de discontinuidades idealmente, proporcionarán los pixels correspondientes a los contornos o fronteras entre las regiones de la imagen. En la práctica este conjunto de pixels no suele caracterizar completamente a esos contornos debido a la presencia de ruido, ruptura en los propios contornos debido a la iluminación no uniforme y otros efectos que introducen espúreos en las discontinuidades de la intensidad. En general, después de los

procedimientos de detección de bordes se suele emplear técnicas de enlazado u otras técnicas de detección de contornos designadas para unir los pixels de bordes en contornos significativos.

3.2. Procesado Local

Uno de los procedimientos más simples para enlazar puntos de bordes es analizar las características de los pixels en un vecindario pequeño (ventana de 3×3) alrededor de cada punto (x, y) donde se ha detectado la presencia de un borde. Todos los puntos que son similares se enlazan, dando lugar a un contorno o frontera de pixels que comparten ciertas propiedades en común. Las dos propiedades que se suelen utilizar son la magnitud del operador de gradiente que ha dado lugar al pixel de borde y la dirección del gradiente en ese punto. Un pixel de borde con coordenadas (x', y') en el vecindario predefinido para el pixel (x, y) va a ser similar en magnitud al pixel (x, y) si

$$|\nabla f(x, y) - \nabla f(x', y')| \leq T, \quad (14)$$

donde T es un cierto umbral en amplitud.

La dirección del vector gradiente venía dado por la ecuación (6). Un pixel de borde con coordenadas (x', y') en el vecindario predefinido para el pixel (x, y) va a ser similar en ángulo al pixel (x, y) si

$$|\alpha(x, y) - \alpha(x', y')| \leq A, \quad (15)$$

donde A es un cierto umbral angular. Hay que tener en cuenta que la dirección del borde en el punto (x, y) es perpendicular a la dirección del gradiente en ese punto. Sin embargo, de cara a comparar las direcciones, la ecuación (15) es equivalente.

Se enlazará un punto, en el vecindario predefinido para (x, y) , con el pixel (x, y) si se cumplen simultáneamente las condiciones de magnitud y de ángulo. Este procedimiento se repetirá para todos los pixels de borde de la imagen. Tras encontrar un pixel de borde que enlaza con el (x, y) se procede a probar con este nuevo pixel de borde, hasta que no se puedan enlazar más pixels de borde. Todos estos pixels enlazados se etiquetan con cierto valor de gris y se procederá a enlazar los restantes pixels de borde con otra etiqueta. Al final habrá tantas etiquetas como conjuntos de pixels enlazados, y que corresponderán a los contornos o fronteras correspondientes a todos los objetos detectados de la imagen.

3.3. Procesado Global empleando la Transformada de Hough

En esta sección vamos a enlazar aquellos puntos de borde que caigan sobre una determinada curva con determinada forma. A diferencia del método de análisis local explicado, ahora se van a considerar relaciones globales entre pixels de borde.

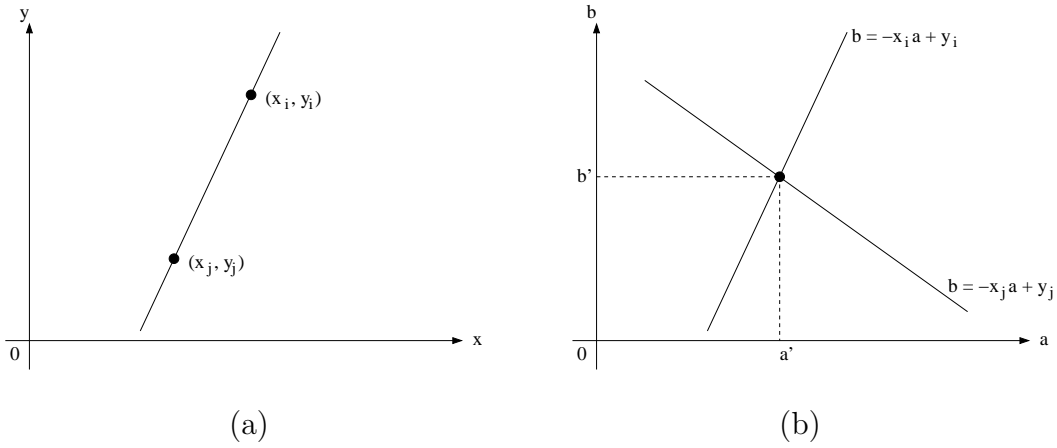


Figura 10: (a) Plano xy . (b) Espacio parámetro.

Supongamos que para n puntos de la imagen, se desean encontrar aquellos subconjuntos de puntos que caen en líneas rectas. Una posible solución podría ser en primer lugar encontrar todas las líneas determinadas por cada par de puntos y entonces encontrar todos los subconjuntos de puntos que están cerca de cada recta en particular. Este problema así planteado requiere encontrar $n(n-1)/2 \sim n^2$ rectas y realizar $n(n(n-1))/2 \sim n^3$ comparaciones de cada punto a cada línea. Este método será inviable salvo en casos triviales.

Una alternativa al método de búsqueda exhaustiva es la transformada de Hough. Consideramos un punto (x_i, y_i) y la ecuación de la recta, de la forma pendiente y ordenada en el origen,

$$y_i = ax_i + b. \quad (16)$$

Por el punto (x_i, y_i) pasan infinitas rectas, pero todas satisfacen la ecuación anterior para diferentes valores de a y b . Sin embargo, escribiendo esta ecuación en la forma

$$b = -x_i a + y_i, \quad (17)$$

y considerando el plano ab (también denominado espacio parámetro) da lugar a una única recta para el par (x_i, y_i) constante. Si ahora consideramos un segundo punto (x_j, y_j) , también va a tener su recta asociada en el espacio parámetro. Estas dos rectas se cortarán en el espacio parámetro en un punto (a', b') , donde a' es la pendiente y b' la ordenada en el origen de la recta que contiene a los puntos (x_i, y_i) y (x_j, y_j) en el plano xy , como se puede ver en la figura 10. De hecho todos los puntos de esa recta en el plano xy darán lugar a rectas diferentes en el espacio parámetro que se cortan en un único punto (a', b') .

El atractivo de la transformada de Hough proviene de subdividir el espacio parámetro en celdas acumuladoras, como se puede ver en la figura 11, donde $(a_{\text{mín}}, a_{\text{máx}})$ y $(b_{\text{mín}}, b_{\text{máx}})$ son los rangos esperados para la pendiente y la ordenada en el origen. La celda de coordenadas (i, j) , con un valor de acumulador $A(i, j)$ corresponde al cuadrado asociado con las coordenadas (a_i, b_j) del espacio parámetro. Inicialmente se ponen todos los acumuladores a cero. Entonces

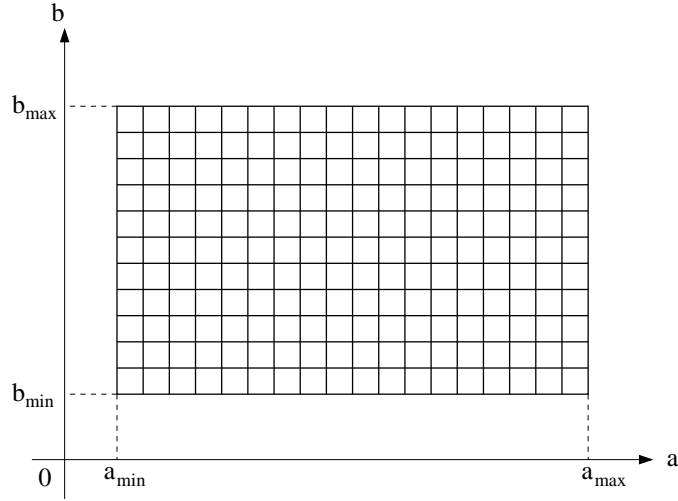


Figura 11: Celdas acumuladoras en el espacio parámetro.

para cada punto (x_k, y_k) de la imagen, permitimos que el parámetro a pueda tomar cualquier valor de entre los a_i permitidos y calculamos b usando la ecuación (17). Los valores resultantes para el parámetro b se redondean hasta los b_j permitidos. Si para un valor a_p resultó un valor b_q se tiene que

$$A(p, q) = A(p, q) + 1. \quad (18)$$

Al final, un valor de M en el acumulador $A(i, j)$ significa que M puntos del plano xy caen sobre la recta $y = a_i x + b_j$. La precisión en la colinealidad de estos puntos dependen del número de celdas del espacio parámetro.

Si subdividimos el eje a en K celdas, para cada punto (x_k, y_k) , obtenemos K valores de b correspondientes a los K posibles valores de A . Si la imagen tiene n puntos, la carga computacional es del orden de nK . La transformada de Hough es lineal en n , y el producto nK es mucho menor que si hubiéramos empleado una búsqueda exhaustiva, a menos que K sea del orden o exceda a n .

Un problema que surge al emplear la ecuación de la recta $y = ax + b$ para representar una línea es que tanto la pendiente como la ordenada en el origen pueden llegar a valer infinito, según la línea se hace vertical. Una forma de solventar este problema consiste en utilizar la representación normal de la recta

$$x \cos \theta + y \sin \theta = \rho. \quad (19)$$

En la figura 12 se puede ver el significado de los nuevos parámetros (ρ, θ) . El uso de esta representación para construir la tabla de acumuladores es similar al método explicado para las rectas en la forma pendiente y ordenada en el origen. A cada punto del plano xy corresponde ahora una senoide en el plano $\rho\theta$ en lugar de una recta. Al igual que antes, M puntos colineales a la recta $x \cos \theta_j + y \sin \theta_j = \rho_i$ darán lugar a M sinusoides que se cortan en el punto (ρ_i, θ_j) en el espacio de parámetros. Incrementando θ y calculando ρ , obtendremos M entradas en el acumulador $A(i, j)$ correspondiente al par (ρ_i, θ_j) .

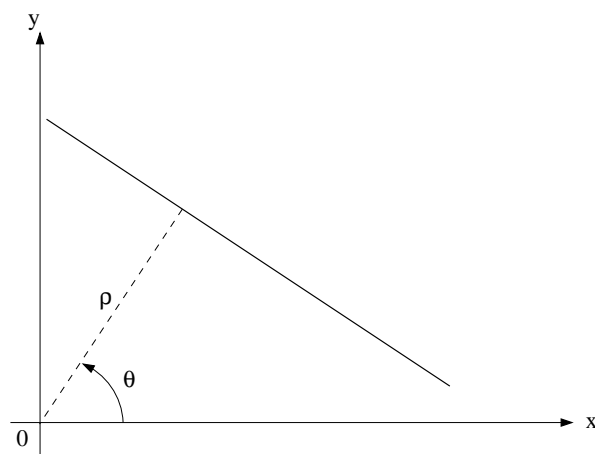


Figura 12: Representación normal de una recta.

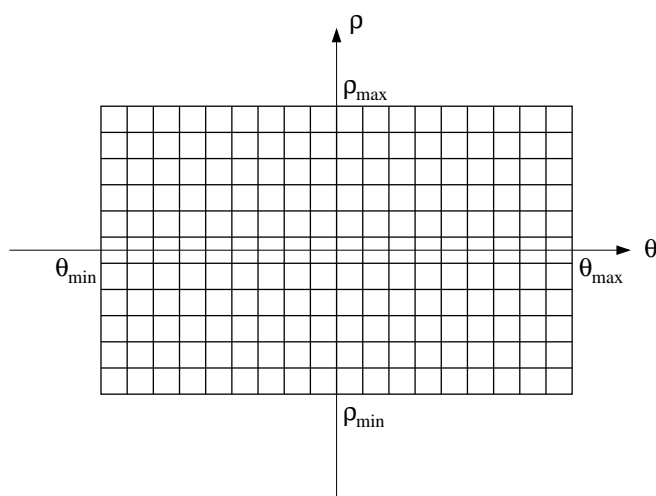


Figura 13: Celdas acumuladoras en el espacio $\rho\theta$.

En la figura 13 podemos ver la tabla de acumuladores del espacio de parámetros en este caso. El rango para el ángulo θ es $\pm 90^\circ$, medido con respecto al eje de abscisas. Se permiten valores negativos de ρ para rectas por detrás del origen de coordenadas del plano xy . Por ejemplo, una recta horizontal corresponde a un valor de $\theta = 0^\circ$ y un valor de ρ igual a la ordenada en el origen, mientras que una recta vertical corresponde a un valor de $\theta = 90^\circ$ y un valor de ρ igual a la abscisa en el origen.

En la figura 14 se ilustra con un ejemplo como funciona la transformada de Hough. La imagen está formada por cuatro puntos correspondientes a las esquinas de un cuadrado. Estos cuatro puntos dan lugar a cuatro sinusoides en el espacio $\rho\theta$. Las cuatro sinusoides se cortan en seis puntos (en la figura 14(b) aparecen ocho puntos, pero hay que recordar que los dos puntos para $\theta = 90^\circ$ son los mismos que los puntos para $\theta = -90^\circ$, por lo que de ocho son seis distintos), correspondientes a las seis rectas posibles que pasan por los cuatro puntos del plano xy , que son a saber, los cuatro lados del cuadrado y

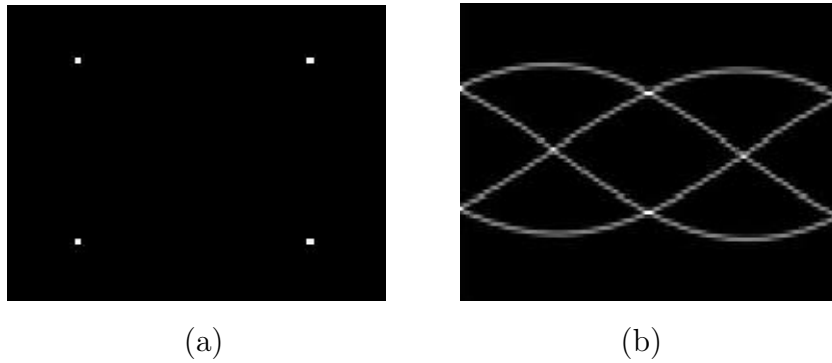


Figura 14: (a) Imagen con cuatro puntos de bordes. (b) Transformada de Hough mostrando seis puntos de corte correspondientes a las seis rectas que pueden pasar por los cuatro puntos.

las dos diagonales.

Aunque hemos hecho un análisis para el caso de rectas, la transformada de Hough también es aplicable a cualquier función de la forma

$$g(\mathbf{v}, \mathbf{c}) = 0, \quad (20)$$

donde \mathbf{v} es un vector de coordenadas y \mathbf{c} es un vector de coeficientes. Por ejemplo, puntos que caen en el círculo

$$(x - c_1)^2 + (y + c_2)^2 = c_3^2, \quad (21)$$

se pueden detectar empleando también la transformada de Hough. En este caso tenemos tres parámetros (c_1, c_2, c_3) , lo que dará lugar a un espacio de parámetros de tres dimensiones, con celdas con forma de cubo y acumuladores de la forma $A(i, j, k)$. El procedimiento en este caso es para cada punto del plano xy , para cada c_1 y para cada c_2 calcular el valor de c_3 y actualizar el acumulador correspondiente a (c_1, c_2, c_3) . Claramente la complejidad de la transformada de Hough es claramente dependiente del tamaño del espacio de parámetros.

4. Técnicas de Umbrales

4.1. Fundamentos

Supongamos que el histograma de los niveles de gris de una imagen $I(x, y)$ es el que se muestra en la figura 15(a). La imagen $I(x, y)$ está compuesta de objetos claros sobre fondo oscuro de tal forma que los niveles de gris está agrupados en dos modos predominantes. Una forma de separar los objetos del fondo consiste en seleccionar un umbral T que separe esos modos. Entonces, cualquier punto (x, y) para el que se cumpla que

$$I(x, y) > T, \quad (22)$$

se lo etiqueta como objeto; en otro caso, como fondo. La figura 15(b) muestra el histograma de otra imagen en un caso más general. En este caso el

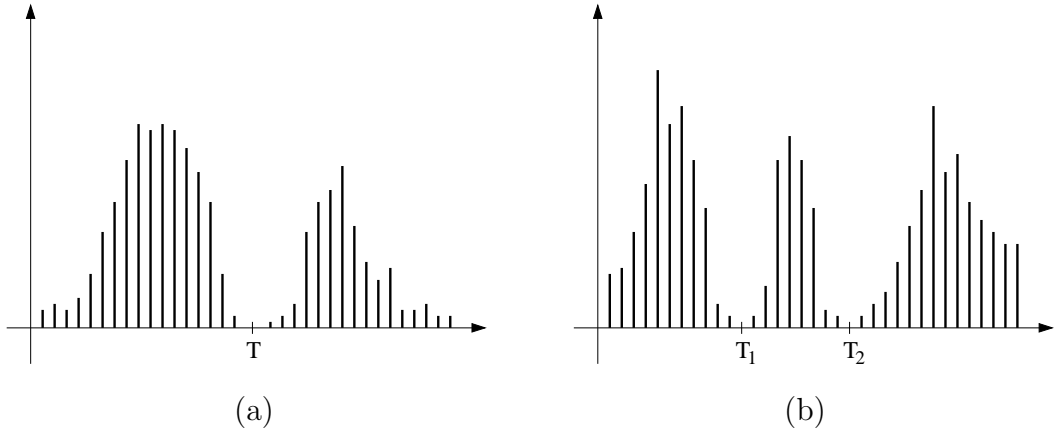


Figura 15: Histogramas de nivel de gris que se pueden segmentar con (a) un único umbral y (b) con múltiples umbrales.

histograma de la imagen está caracterizado por tres modos dominantes. Esto ocurrirá cuando tengamos dos tipos de objetos claros sobre fondo oscuro, por ejemplo. Se puede utilizar el mismo principio para clasificar cada punto (x, y) . Si $T_1 < I(x, y) \leq T_2$ entonces se lo etiqueta como primer objeto, si $I(x, y) > T_2$ como segundo objeto y si $I(x, y) \leq T_1$ como fondo. En general, este tipo de clasificación con varios umbrales es menos viable, ya que es más difícil determinar esos umbrales que aislen de forma efectiva las regiones de interés, especialmente cuando el número de modos del histograma aumenta. En este caso es mejor emplear umbrales variables.

En general, un método de umbral se puede ver como una operación en la que se hace un test de cada pixel con respecto a una función T de la forma

$$T = T(x, y, p(x, y), I(x, y)), \quad (23)$$

donde $I(x, y)$ es el nivel de gris del punto (x, y) y $p(x, y)$ denota cualquier propiedad local de ese punto (como por ejemplo el nivel de gris medio en un vecindario centrado en (x, y)). El método de umbral dará lugar a otra imagen $B(x, y)$ definida por

$$B(x, y) = \begin{cases} 1 & \text{si } I(x, y) > T \\ 0 & \text{si } I(x, y) \leq T. \end{cases} \quad (24)$$

En este caso un pixel con etiqueta 1 de la imagen B corresponderá a objetos, mientras que un pixel con etiqueta 0 corresponderá al fondo.

Cuando T dependa sólo del nivel de gris $I(x, y)$ se denomina umbral *global* (en la figura 15(a) se puede ver un ejemplo en este caso). Si T depende tanto del nivel de gris $I(x, y)$ como de la propiedad local $p(x, y)$, el umbral se denomina *local*. Si, además, T depende de las coordenadas espaciales x e y , el umbral se denomina *dinámico*.

En la figura 16(a) se puede ver el histograma de la imagen de la figura 6(a) para el que se ha fijado un umbral T . En la figura 16(b) se muestra el resultado de segmentar dicha imagen con ese umbral T . Como se puede

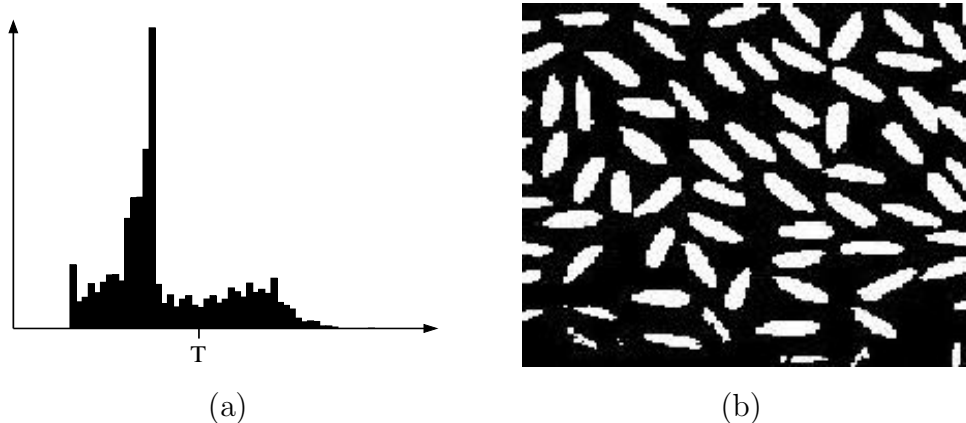


Figura 16: (a) Histograma y umbral T para la imagen de la figura 6(a). (b) Imagen segmentada usando el umbral T .

apreciar el histograma de esta imagen es bimodal. Sin embargo, el valle no está claramente marcado debido fundamentalmente a que la iluminación de la imagen no es constante. Esto influye enormemente en el histograma como veremos a continuación. Sin embargo, en este caso la segmentación es bastante adecuada, excepto en la zona inferior donde la iluminación de la imagen es peor.

4.2. Influencia de la Iluminación

La formación de una imagen $I(x, y)$ se puede ver como

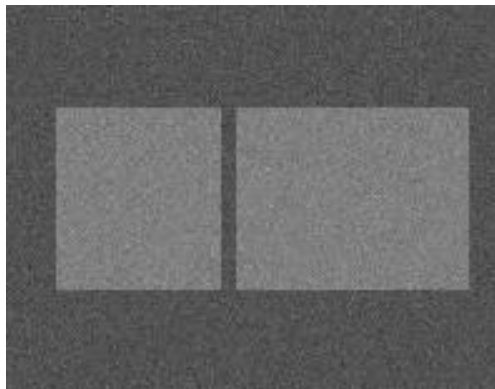
$$I(x, y) = R(x, y)L(x, y), \quad (25)$$

donde $R(x, y)$ es la componente de reflectancia y $L(x, y)$ la de iluminación. Vamos a ver cual es el efecto de la componente de iluminación $L(x, y)$ de cara a poder segmentar la imagen $I(x, y)$.

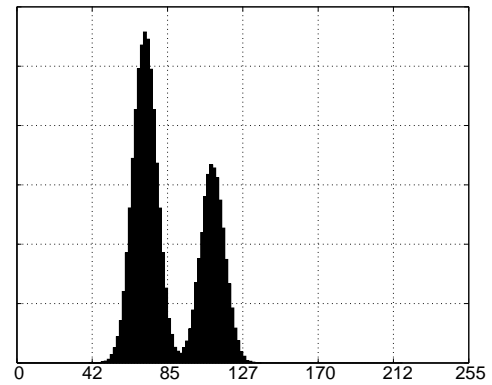
Vamos a considerar la función de reflectancia $R(x, y)$ generada con ordenador de la figura 17(a). El histograma de esta función, como se aprecia en la figura 17(b), es claramente bimodal y la imagen se puede segmentar fácilmente mediante un umbral situado en el valle del histograma. Consideramos ahora la función de iluminación $L(x, y)$ generada con ordenador de la figura 17(c). Multiplicando las funciones de reflectancia e iluminación obtenemos la imagen $I(x, y)$ mostrada en la figura 17(d). En el histograma de la imagen $I(x, y)$, como se puede ver en la figura 17(e), el valle prácticamente ha desaparecido por completo, por lo que la segmentación mediante un único umbral es en este caso imposible. Aunque en general casi nunca se dispone de la función de reflectancia aislada, este ejemplo muestra que la naturaleza reflectiva de los objetos y del fondo son tales que son fácilmente separables. Sin embargo, la imagen resultante, debido a una mala iluminación, es muy difícil de segmentar.

La razón por la que el histograma de la reflectancia se ve modificado por la iluminación se puede explicar de forma sencilla. Vamos a tomar logaritmos en la ecuación (25) obteniendo otra imagen

$$Z(x, y) = \ln I(x, y) = \ln R(x, y) + \ln L(x, y) = R'(x, y) + L'(x, y). \quad (26)$$



(a)



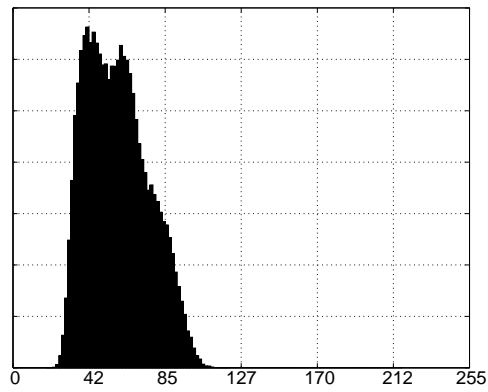
(b)



(c)



(d)



(e)

Figura 17: (a) Función de reflectancia $R(x, y)$ generada con ordenador. (b) Histograma de la función de reflectancia. (c) Función de iluminación $L(x, y)$ generada con ordenador. (d) Imagen $I(x, y)$ producto de las funciones de reflectancia e iluminación. (e) Histograma de la imagen.

Si $R'(x, y)$ y $L'(x, y)$ son variables independientes, el histograma de $Z(x, y)$ viene dado por la convolución de los histogramas de $R'(x, y)$ y $L'(x, y)$. Si la iluminación $L(x, y)$ es constante, $L'(x, y)$ también lo es, y su histograma será un impulso. La convolución de este impulso con el histograma de la reflectancia $R'(x, y)$ no lo modifica y por lo tanto la imagen $I(x, y)$ tiene el mismo histograma que la reflectancia $R(x, y)$. Si por el contrario la iluminación $L'(x, y)$ tiene un histograma más ancho, el proceso de la convolución con el histograma de la reflectancia $R'(x, y)$, dará lugar a que el histograma de la imagen $I(x, y)$ sea sustancialmente diferente al histograma de la componente de reflectancia $R(x, y)$. El grado de distorsión dependerá de cómo de ancho sea el histograma de la componente de iluminación $L(x, y)$, es decir, de cómo de no uniforme sea la iluminación.

Cuando se tenga acceso a la fuente de iluminación, una solución que se emplea habitualmente para compensar la no uniformidad de ésta, consiste en proyectar la iluminación sobre una superficie clara y plana. Esta proyección dará lugar a una imagen $G(x, y) = kL(x, y)$, donde k es una constante que depende de la reflectividad de la superficie. Ahora para cualquier otra imagen $I(x, y)$ formada a partir de la ecuación (25) para una función de reflectividad dada $R(x, y)$ y para la misma función de iluminación $L(x, y)$, podemos compensar la iluminación haciendo

$$H(x, y) = \frac{I(x, y)}{G(x, y)} = \frac{R(x, y)}{k}. \quad (27)$$

Entonces si $R(x, y)$ se puede segmentar empleando un umbral T , entonces la imagen $H(x, y)$ también, empleando un umbral T/k . Este método se puede emplear siempre que la iluminación no cambie de imagen a imagen.

4.3. Umbral Óptimo

Supongamos que la imagen contiene únicamente dos niveles de gris predominantes. El histograma de esta imagen se puede considerar una estimación de la función densidad de probabilidad de nivel de gris, $p(z)$. Esta función será la mezcla de dos funciones unimodales, una para la región clara y otra para la región oscura de la imagen. Además, los pesos de estas densidades van a ser proporcionales al área de cada modo. Si se conoce o se supone la forma de las funciones de densidad de cada modo, se puede determinar el umbral óptimo (en términos de mínimo error) para la imagen a segmentar.

Vamos a suponer que la imagen contiene dos valores de nivel de gris combinados con ruido aditivo Gaussiano. La función densidad de probabilidad de primer orden del nivel de gris de la imagen vendrá dado por

$$p(z) = P_1 p_1(z) + P_2 p_2(z), \quad (28)$$

donde, en el caso Gaussiano se tiene

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(z - \mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(z - \mu_2)^2}{2\sigma_2^2}\right], \quad (29)$$

donde μ_1 y μ_2 son los valores medios de nivel de gris, σ_1 y σ_2 las desviaciones estándar en torno a la media y P_1 y P_2 las probabilidades *a priori* de los dos niveles de gris. Se debe satisfacer la restricción

$$P_1 + P_2 = 1, \quad (30)$$

de forma, que la densidad $p(z)$ tiene cinco parámetros desconocidos. Si se conocen estos cinco parámetros se puede determinar el umbral óptimo de forma sencilla.

Supongamos que la región oscura corresponde al fondo de la imagen, y las regiones claras a objetos. En este caso $\mu_1 < \mu_2$ y se debe definir un umbral T de forma que los pixels con nivel de gris menores que T se puedan considerar como pixels de fondo y los pixels con nivel de gris mayores que T pixel objeto.

La probabilidad de clasificar erróneamente un pixel de objeto como de fondo es

$$E_1(T) = \int_{-\infty}^T p_2(z) dz. \quad (31)$$

De forma similar, la probabilidad de clasificar erróneamente un pixel de fondo como objeto es

$$E_2(T) = \int_T^{\infty} p_1(z) dz. \quad (32)$$

Entonces la probabilidad total de error es

$$E(T) = P_2 E_1(T) + P_1 E_2(T). \quad (33)$$

Para encontrar el valor de T que da lugar a una probabilidad de error de clasificación mínima, podemos derivar la ecuación (33) con respecto a T e igualar a cero. Entonces

$$P_1 p_1(T) = P_2 p_2(T). \quad (34)$$

En el caso Gaussiano, tomando logaritmos, la ecuación (34) resulta en la ecuación de segundo grado

$$AT^2 + BT + C = 0, \quad (35)$$

donde

$$A = \sigma_1^2 - \sigma_2^2 \quad (36)$$

$$B = 2(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2) \quad (37)$$

$$C = \sigma_1^2 \mu_2^2 - \sigma_2^2 \mu_1^2 + 2\sigma_1^2 \sigma_2^2 \ln \frac{\sigma_2 P_1}{\sigma_1 P_2}. \quad (38)$$

De las dos soluciones una corresponderá a un máximo local y otra a un mínimo local. Es necesario sustituir ambos resultados en la ecuación (33) y descartar la que de lugar a un error mayor, quedándonos con la otra solución como la válida.

En el caso de que las varianzas sean iguales, $\sigma^2 = \sigma_1^2 = \sigma_2^2$, la ecuación (35) tiene una única solución

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln \frac{P_2}{P_1}, \quad (39)$$

que corresponde en este caso al error mínimo. Si las probabilidades *a priori* son iguales, $P_1 = P_2$, el umbral óptimo viene dado por el valor medio de las medias. Esto mismo se cumple para el caso $\sigma = 0$.

En el caso de que las densidades de probabilidad sigan otras leyes conocidas, como la Rayleigh o la log-normal, se puede aplicar este mismo procedimiento para la determinación del umbral óptimo.

4.4. Umbrales basados en Varias Variables

Hasta ahora hemos considerado la determinación de umbrales para una única variable: el nivel de gris de la imagen. En algunos casos, se dispone de varias variables que caracterizan cada pixel de la imagen. Un ejemplo muy claro son las imágenes en color, donde se dispone de las componentes RGB para formar la imagen compuesta en color. En este caso, cada pixel se caracteriza por tres variables, y se puede construir un histograma en tres dimensiones.

El concepto de segmentación mediante umbral en este caso da lugar el empleo de técnicas de clasificación o *clustering*. Si se han encontrado K *clusters* significativos en el espacio de las variables, la imagen se puede segmentar asignando a cada pixel una de las K etiquetas a todos aquellos pixels cercanos al *cluster* correspondiente a esa etiqueta. La complejidad de esta clasificación depende del número de variables y del número de *clusters*. Además si se desconoce el número de *clusters* el problema se complica aun más.

En el caso particular que se conozca el número de *clusters* se puede emplear el algoritmo LBG o *K-means*. Este algoritmo permite determinar iterativamente el centroide de cada uno de los K *clusters* y la partición del espacio de los parámetros en K zonas, una para cada *cluster*. Partiendo de una estimación inicial para los centroides, se clasifica los datos según esos centroides. A partir de los datos clasificados para cada *cluster* se vuelve a determinar el centroide para ese *cluster*. Esto se repite hasta que la posición de los centroides de los *clusters* y la clasificación de los datos no cambie apreciablemente. La imagen segmentada viene dada entonces por las etiquetas correspondientes a la última clasificación.

Es particularmente importante la clasificación de imágenes en color basada en tono y saturación. Estas propiedades de color son particularmente importantes, a diferencia de los canales RGB, porque son equivalentes a como se perciben los colores en el sistema visual humano. En este caso tenemos dos variables y se pueden emplear técnicas de clasificación o *clustering* en dos dimensiones. En la figura 18 se puede ver un ejemplo para tres *clusters* junto con las regiones de clasificación (umbrales en dos dimensiones).

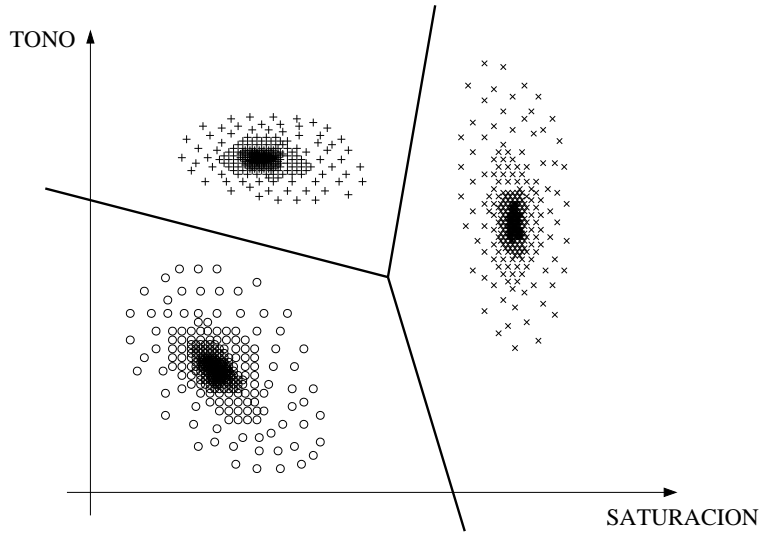


Figura 18: Ejemplo de *clustering* para dos variables y tres *clusters*.

5. Segmentación Orientada a Regiones

5.1. Preliminares

Sea R la región correspondiente a la imagen a segmentar. Vamos a ver el proceso de segmentación como un proceso en el cual dividimos la región R en n subregiones R_1, R_2, \dots, R_n , tal que

$$\bigcup_{i=1}^n R_i = R \quad (40)$$

$$R_i \text{ es una región conectada, } i = 1, 2, \dots, n \quad (41)$$

$$R_i \cap R_j = \emptyset \text{ para todo } i \text{ y } j \text{ con } j \neq i \quad (42)$$

$$P(R_i) = \text{CIERTO para } i = 1, 2, \dots, n \quad (43)$$

$$P(R_i \cup R_j) = \text{FALSO para todo } i \text{ y } j \text{ adyacentes con } j \neq i, \quad (44)$$

donde $P(\cdot)$ es un predicado lógico definido sobre la partición de R y \emptyset es el conjunto vacío.

La ecuación (40) indica que la segmentación debe ser completa, es decir, que todo pixel debe estar en una región. La ecuación (41) requiere que todos los pixels pertenecientes a una región estén conectados. La ecuación (42) lleva consigo regiones disjuntas, por eso mismo, los conjuntos de pixels R_1, R_2, \dots, R_n son una partición de R . La ecuación (43) se refiere a la propiedad que deben cumplir los pixels dentro de cada región segmentada (por ejemplo $P(R_i) = \text{CIERTO}$ si todos los pixels de R_i tienen el mismo nivel de gris dentro de una cierta tolerancia). Finalmente, la ecuación (44) indica que las regiones adyacentes R_i y R_j deben ser distintas con respecto a la propiedad P .

5.2. Crecimiento de Regiones

Como el nombre indica, el crecimiento de regiones es un procedimiento mediante el cual se agrupan pixels o subregiones en regiones mayores. El procedimiento más sencillo se denomina agregación de pixels, que comienza a partir de un conjunto de pixels semilla, de forma que a partir de cada semilla se crecen regiones añadiendo pixels a dicha semilla de entre aquellos pixels vecinos que tienen propiedades similares. El resultado de la segmentación dará lugar como mucho a tantas regiones como semillas haya. Sin embargo, puede darse el caso de que dos de esas semillas correspondan a pixels de la misma región. En este caso el crecimiento desde una de las semillas absorberá a la otra, que en este caso deberá ser descartada. Un ejemplo sencillo, pero muy usado en la práctica de similitud es la diferencia absoluta en el nivel de gris. Fijado un umbral T se va calculando la diferencia en valor absoluto del nivel de gris del pixel en cuestión (en el vecindario de la región crecida hasta el momento) con respecto al nivel de gris de la semilla y si no se supera ese umbral T se añade a la región.

Dos problemas fundamentales en el crecimiento de regiones son: por un lado, la selección de las semillas o puntos de partida que representen adecuadamente a las regiones de interés; y por otro, la elección de las propiedades adecuadas que permitan ir añadiendo pixels durante el proceso de crecimiento. La selección de los puntos de partida en muchos casos depende de la naturaleza de la imagen a segmentar. Por ejemplo, en aplicaciones militares con imágenes de infrarrojos, los blancos de interés normalmente desprenden calor, por lo que corresponden a pixels claros frente a fondo oscuro. En este caso, los pixels claros son una elección natural para las semillas. En ausencia de conocimiento *a priori* sobre el problema, un procedimiento que suele dar lugar a buenos resultados consiste en calcular para cada pixel de la imagen las mismas propiedades que luego se emplearán durante el proceso de crecimiento. Si el resultado del cálculo de dichas propiedades da lugar a agrupaciones de los valores para dichas propiedades, se pueden emplear como semillas los pixels cercanos a los centroides de dichas agrupaciones. Por ejemplo, los pixels correspondientes a los valores máximos de cada modo del histograma de nivel de gris podrían servir como semillas cuando se emplee como propiedad la similitud en el nivel de gris.

La selección del criterio de similitud depende no sólo del problema considerado, sino también del tipo de imagen disponible. En imágenes en color o multiespectrales (fotografías desde satélite, por ejemplo) se pueden emplear propiedades de similitud en los diferentes canales disponibles. Sin embargo, en general, lo habitual es trabajar con imágenes de intensidad con un único canal disponible. En este caso las propiedades se basan en descriptores locales a partir de una única imagen como son la intensidad y propiedades espaciales (basadas momentos y textura). En cualquier caso siempre hay que tener en cuenta la conectividad durante el proceso de crecimiento para que el resultado tenga significado dentro de su contexto.

Otro problema añadido en el crecimiento de regiones es la regla de parada. Básicamente, el crecimiento termina cuando no existen más pixels en el

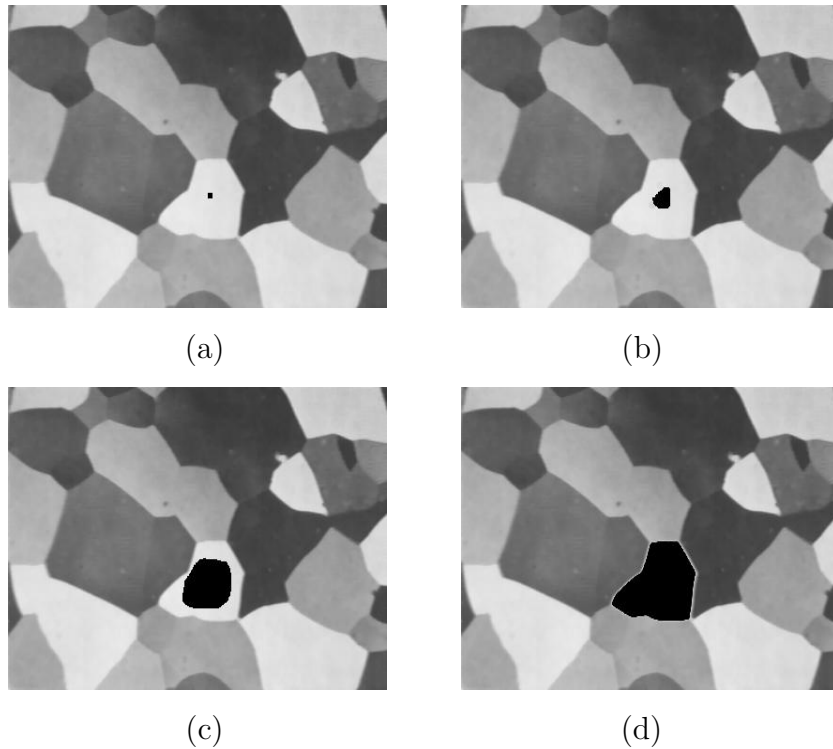


Figura 19: (a) Imagen en la que se ha marcado un punto interior a la región a segmentar. (b)-(d) Proceso de crecimiento de la región marcada.

vecindario de la región ya crecida que cumplan el criterio de similitud. Los criterios de similitud mencionados son locales y no tienen en cuenta la “historia” del crecimiento de la región. Se pueden emplear criterios adicionales que incrementen la potencia del algoritmo de crecimiento de regiones, utilizando el concepto de similitud entre el pixel candidato y los pixels de la región crecida hasta ese momento, o usando el tamaño y la forma de la región crecida. La utilización de estos tipos de descriptores se basa en la suposición de que un modelo de los resultados esperados ya está parcialmente disponible.

En la figura 19(a) se muestra una imagen con varias regiones en la que se muestra una única semilla en el interior de una de las regiones a segmentar. El criterio empleado para el crecimiento es que el valor absoluto de la diferencia entre los niveles de gris del pixel candidato y la semilla no exceda el diez por ciento de la diferencia entre el mayor nivel de gris de la imagen completa y el menor. Además los pixels agregados a la región deben tener conectividad tipo ocho con respecto a los pixels ya incluidos en la región. En la figura 19(b) se muestra el proceso de crecimiento tras unas pocas iteraciones, en la figura 19(c) se muestra el proceso para la mitad de las iteraciones y en la figura 19(d) el resultado del crecimiento de la región para cuando ninguno de los pixels vecinos de la región cumple el criterio de similitud.

5.3. División y Fusión de Regiones

Un proceso alternativo al crecimiento de regiones es subdividir inicialmente la imagen en un conjunto arbitrario de regiones disjuntas y posteriormente

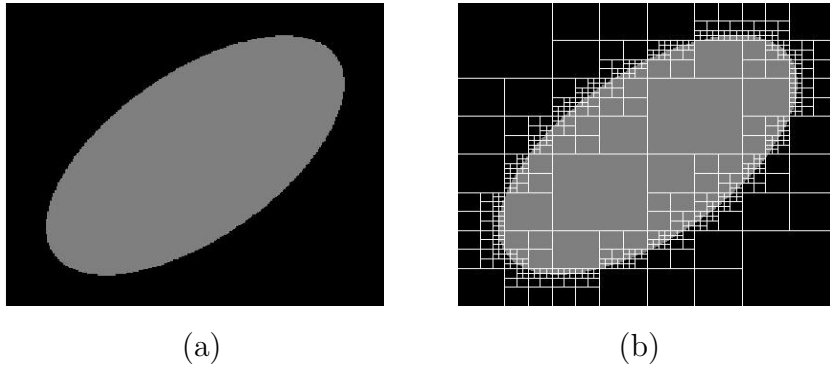


Figura 20: (a) Imagen sintética. (b) División de la imagen correspondiente a la representación *quadtree*.

fusionar y/o dividir estas regiones con el objetivo de que finalmente se cumplan las condiciones dadas por las ecuaciones (40)-(44).

Sea R la región correspondiente a la imagen completa y sea P el predicado de similitud elegido. Un procedimiento para segmentar R consiste en subdividir recursivamente cada región en cuadrantes más y más pequeños hasta que se cumpla para cada región R_i que $P(R_i) = \text{CIERTO}$. Es decir si $P(R) = \text{FALSO}$, se divide la imagen en cuatro regiones (cuadrantes). Si para alguno de los cuadrantes P es FALSO, se vuelve a dividir dicho cuadrante en cuatro subcuadrantes y así sucesivamente. Esta técnica de división de una imagen da lugar a un tipo de representación bastante conveniente denominada *quadtree*, es decir, un árbol de regiones en el que cada región tiene exactamente cuatro descendientes. La raíz del árbol corresponde siempre a la imagen completa R . Cada nodo corresponde a una subdivisión. En la figura 20(a) podemos ver una imagen sintética generada por ordenador de una elipse clara sobre un fondo oscuro. En este caso el criterio de similitud es que la diferencia en valor absoluto en los niveles de gris sea menor que cierto umbral T . En la figura 20(b) podemos ver el resultado de la división de la imagen en cuadrantes correspondientes a los nodos finales del *quadtree*. Las zonas más uniformes dan lugar a regiones mayores, mientras que las zonas de transición dan lugar a regiones pequeñas, de forma que siempre se cumpla el predicado P en cada región.

Tras la etapa de división, la partición final contiene regiones adyacentes con propiedades idénticas, por lo que no se cumple la propiedad correspondiente a la ecuación (44). Esto se puede solucionar permitiendo la fusión de regiones adyacentes cuando la unión de ambas cumplan el predicado P .

Podríamos resumir el procedimiento de división y fusión de regiones como sigue:

1. Dividir una región R_i en sus cuatro cuadrantes disjuntos siempre que $P(R_i) = \text{FALSO}$.
2. Fusionar dos regiones adyacentes R_i y R_j siempre que $P(R_i \cup R_j) = \text{CIERTO}$.
3. Parar el algoritmo cuando no sea posible realizar más divisiones y fusiones.