

# ***Segmentación de imágenes***

***Extracto de “Visión por Computador. Fundamentos y métodos”. Arturo de la Escalera. Prentice Hall, 2001***

Copia para el alumno con fines didácticos

## 7. Segmentación de imágenes.

En cualquier imagen se encontrarán presentes uno o varios objetos localizados en un entorno. El objetivo de la segmentación es separar dichos objetos del medio en el que se encuentran y distinguirlos entre sí. Para ello utilizará alguna de las propiedades o características vistas hasta ahora: niveles de gris, color, textura, bordes o movimiento. Después de esta separación estará el proceso de descripción individualizada de los objetos encontrados y su reconocimiento e interpretación.

La segmentación se basa en tres propiedades:

- Similitud. Cada uno de los píxeles de un elemento tiene valores parecidos para alguna propiedad.
- Discontinuidad. Los objetos destacan del entorno y tienen por tanto unos bordes definidos.
- Conectividad. Los píxeles pertenecientes al mismo objeto tienen que ser contiguos, es decir, deben estar agrupados.

Estas suposiciones no son fáciles de cumplir y sin embargo son fundamentales para una buena segmentación. Respecto a la primera los objetos deberían presentar una apariencia uniforme, lo que no va a ser posible al no ser la iluminación constante, existir pequeñas variaciones en el material, la diferencia entre las ganancias para cada píxel de la cámara, la existencia de brillos o la presencia de ruido en la imagen. Respecto a la segunda condición los bordes no van a estar siempre bien definidos. Por último se pueden dar el caso de ocultamientos parciales de un objeto por parte de otro lo que dificultará el cumplimiento de la conectividad de todos los puntos de los objetos. Parece bueno volver a insistir aquí en que la visión por computador se encuentra lejos de poder analizar imágenes complejas y que a menudo se tienen que realizar fuertes simplificaciones. De hecho, la importancia de la iluminación en las aplicaciones industriales radica en que con ella se puede simplificar notablemente la etapa de segmentación. Esta limitación, sin embargo, no es un impedimento para que el análisis de imágenes por ordenador tenga ahora mismo un gran rango de aplicaciones; mayor incluso del que en la actualidad se está aplicando.

Las técnicas para obtener la segmentación se basan por tanto en la búsqueda de las partes uniformes de la imagen o justo lo contrario, aquellas partes donde se produce un cambio. Dependerá del caso concreto el que se siga un método u otro o la conjunción de ambos. Después de haber detectado los puntos que presentan esa discontinuidad (bordes) habrá que encontrar un camino entre el píxel  $P_1$  y el píxel  $P_N$ . Un camino es una secuencia de puntos  $P_2, P_3, \dots, P_{N-1}$ , donde el píxel  $P_{i+1}$  es vecino del píxel  $P_i$ . Se dirá que es un camino sencillo si ningún punto aparece más de una vez y ningún píxel tienen más de dos vecinos. Un camino cerrado es un camino sencillo en el que el primer punto es vecino del último. Una región es un conjunto de píxeles en los que existe un camino entre cualquier pareja de sus puntos, y todos los píxeles del camino pertenecen a la región.

Después de la segmentación se tienen una serie de regiones:

- $R_i$ : Son las regiones que no tocan los bordes de la imagen.
- $R$ : Es el conjunto unión de todas las regiones  $R_i$
- $R_C$ : Es el conjunto complemento de  $R$ .

Entonces se define:

- Fondo: son los puntos que pertenecen a  $R_C$  y son continuos a los bordes de la imagen.
- Agujeros: son los puntos que pertenecen a  $R_C$  y no son continuos a los bordes de la imagen. Los agujeros son puntos que no pertenecen al objeto pero que están rodeados de él.

## 7.1. Segmentación basada en la detección de bordes.

La segmentación basada en la detección de bordes utiliza la información suministrada por los detectores explicados en el capítulo sexto. Como ya se comentó, estos métodos funcionan peor cuando varios objetos están en contacto, semiocultos por otros o por discontinuidades en los bordes producto del ruido. Es por ello que los puntos detectados no constituirán todos los píxeles de los bordes reales, presentándose interrupciones y apareciendo puntos que no pertenecen a ningún borde. Por ello después del algoritmo de detección se aplican otros que analizando los valores detectados determinan las fronteras entre objetos.

Los inconvenientes de estos métodos serán:

- Que al intentar eliminar los efectos del ruido también eliminan los bordes de objetos pequeños, ya que se basan en el pequeño número de píxeles que forman el borde.
- Que suelen asimilar los puntos detectados a curvas de forma conocida que serán siempre una aproximación de los bordes reales.

### 7.1.1 Método basados en el gradiente.

El método del gradiente se basa en los operadores vistos con anterioridad: Roberts por un lado y Sobel, Prewitt y el Isotrópico por otro. Con ellos puede obtenerse el módulo y el argumento del gradiente.

Si se aplica Roberts el módulo será:

$$|G| = \sqrt{(a_4 - a_8)^2 + (a_7 - a_5)^2}$$

y el ángulo:

$$\angle G = -\frac{\pi}{4} + \arctan \frac{a_7 - a_5}{a_4 - a_8}$$

Según Sobel, Prewitt, o el Isotrópico el módulo y el ángulo se obtienen con las fórmulas:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

$$\angle G = \arctan \frac{G_y}{G_x}$$

El siguiente paso es agrupar los píxeles pertenecientes a los bordes con sus vecinos. Para ello se aplican las definiciones de conectividad explicadas con anterioridad. Para que dos píxeles pertenezcan al mismo borde deben cumplir que:

- Estar conectados con conectividad cuatro u ocho.
- Y que la diferencia entre el módulo y la dirección de los dos gradientes sean menores que dos valores prefijados:

$$|G_1 - G_2| \leq T_G$$

$$|\theta_1 - \theta_2| \leq T_\theta$$

Un ejemplo puede verse en la figura 7.1. En ella se tiene el objeto que se quiere detectar. Primero se obtendrían el gradiente vertical y horizontal, para obtener después su módulo y argumento. Eligiendo un punto al azar y aplicando las dos condiciones anteriores se obtendrían una serie de segmentos. Dependiendo de lo que interese el problema estaría resuelto o se analizarían los segmentos para ver si constituyen un camino cerrado.

Sin embargo en casos más complicados los dos criterios anteriores no serán del todo válidos. Se tendrán píxeles como pertenecientes a algún borde cuando son producto del ruido y también se presentarán discontinuidades en los bordes verdaderos. Por ello a la hora de determinar que puntos determinan el borde se analizará primero que píxeles están conectados (decidiendo si se seguirá conectividad de orden cuatro u ocho). Con esto se tendrá generalmente un camino principal del que salen varias ramas por lo que habrá varios caminos posibles. Se tendrá por tanto que buscar el vecino que cumpla ciertas condiciones de forma mínima.

La teoría de grafos ayuda al análisis de los bordes. Un grafo es un conjunto de nodos y arcos entre ellos. Por ello se puede tomar los puntos de borde como nodos dentro de un grafo (Nilson), definiéndose una función de coste entre los puntos inicial y final  $x_1$  y  $x_N$ . Por ejemplo la función:

$$C = -\sum_{i=1}^N |e(x_i)| + a \sum_{i=2}^N |\theta(x_i) - \theta(x_{i-1})| + b \sum_{i=2}^N |e(x_i) - e(x_{i-1})|$$

representa el coste desde el punto inicial a otro intermedio y desde este al final. Así cuanto mayor sea el módulo del gradiente menor será el coste, mientras que la diferencia entre los módulos y dirección del gradiente penalizan el coste. Este coste se calcularía para cada uno de los posibles caminos y se elegiría aquel que su coste sea menor. Aunque los resultados son buenos es complicado de implementar ya que se tienen que tomar en cuenta muchos elementos (todos lo posibles caminos –el principal y las ramas -, bucles infinitos, etc.), además puede darse el caso de que cuando más corto sea el camino su coste pueda ser menor, por último se necesita saber el punto inicial y el final, lo cual no es siempre posible.

Como alternativa se tiene la aplicación de la programación dinámica (*dynamic programming*). La programación dinámica busca el valor óptimo de una función en las que no todas las variables están relacionadas simultáneamente. En el presente caso supone que el camino óptimo entre dos puntos tiene subcaminos óptimos para los puntos intermedios (Bellman y Dreyfus). Así, se podría definir una función de coste similar a la anterior (solo que ahora buscando el camino que obtenga el valor más alto):

$$C = \sum_{i=1}^N |e(x_i)| - a \sum_{i=2}^N |\theta(x_i) - \theta(x_{i-1})|$$

Función que puede expresarse de forma recursiva:

$$C(x_1, x_N) = C(x_1, x_{N-1}) + |e(x_N)| - a |\theta(x_N) - \theta(x_{N-1})|$$

Entonces el camino óptimo puede dividirse en dos subcaminos óptimos que satisfagan:

$$C_{\text{optimo}}(x_1, x_N) = \max C(x_1, x_N) = \max \{C(x_1, x_{N-1}) + |e(x_N)| - a |\theta(x_N) - \theta(x_{N-1})|\}$$

$$C_{\text{optimo}}(x_1, x_N) = C_{\text{optimo}}(x_1, x_{N-1}) + \max \{|e(x_N)| - a |\theta(x_N) - \theta(x_{N-1})|\}$$

Luego a partir del primer píxel se buscará cada píxel que maximice el segundo sumando. Así en el caso de la figura 7.2 se tienen tres píxeles que pueden ser el punto inicial del camino ( $a, b$  y  $c$ ), otros tres que pueden ser el punto final ( $g, h, i$ ) y los píxeles

intermedios ( $d, e, y f$ ); además se representa el valor de pasar de uno a otro. Si se quiere pasar por  $d$  el camino óptimo sería  $a-d$  al tener el valor mayor, para el punto  $e$  es  $b-e$  y para el punto  $f$  es  $c-f$ . Con esto se tendría los costes asociados a los puntos intermedios:

$$C(a,d)=7$$

$$C(b,e)=6$$

$$C(c,f)=5$$

Para pasar a los puntos finales se vería que para llegar a  $g$  el camino es  $d-g$ , para  $h$ ,  $d-h$  y para  $i$  es  $f-i$ , con los costes:

$$C(d,g)=5$$

$$C(d,h)=6$$

$$C(f,i)=6$$

Luego el coste total es:

$$C(a,g)=C(a,d)+C(d,g)=7+5=12$$

$$C(a,h)=C(a,d)+C(d,h)=7+6=13$$

$$C(c,i)=C(c,f)+C(f,i)=5+6=11$$

Por lo que el camino a seguir pasa por los puntos  $a-d-h$

### 7.1.2. La transformada de Hough.

Los métodos descritos anteriormente tienen en cuenta solamente la información local del entorno de vecindad de cada píxel. Existen otros métodos que se basan en la información que suministra toda la imagen, siendo la transformada de Hough el más importante de ellos. Esta transformada busca formas geométricas en toda la imagen encontrando los parámetros de aquella que contenga a más puntos en ella.

*La transformada de Hough para rectas.*

Para evitar las singularidades que pueden dar las rectas de pendiente infinita, la ecuación de una línea recta puede escribirse como:

$$x \cos \theta + y \sin \theta = \rho$$

quedando por tanto cada recta definida por su par de parámetros  $\theta$  y  $\rho$  (figura 7.3). Todos los puntos que pertenezcan a la misma recta coincidirán en las componentes  $\theta$  y  $\rho$ .

Por tanto se pasaría primero un detector de bordes y se consideraría como puntos pertenecientes a él aquellos cuyo valor sea más alto de un valor prefijado. Para cada punto  $(x_i, y_i)$  detectado se calcula la curva:

$$x_i \cos \theta + y_i \sin \theta = \rho$$

donde ahora  $\theta$  y  $\rho$  son variables y que representa todas las rectas que pasan por el punto  $(x_i, y_i)$ . Para dos puntos cualesquiera (figura 7.4) la intersección de sus dos curvas senoidales estará en aquellos valores  $(\theta_i, \rho_i)$  que definen la recta que los contiene. Al final aquellos valores más repetidos coinciden con la recta que contiene al mayor número de píxeles. Se tiene así la figura 7.5 donde está la imagen (a) sobre la que se quieren detectar las rectas presentes. Se obtendría los bordes (b) de la imagen y se calcularían todas las posibles rectas (c) según la fórmulas antes vista. Puede observarse como existen unos valores donde se cortan gran parte de ellas. Obteniendo los valores de  $\theta$  y  $\rho$  se representan las rectas detectadas de forma superpuesta a la imagen original (d).

Un inconveniente de la transformada de Hough es su coste computacional ya que el número de operaciones es alto. Aunque este coste cada vez es menor debido al desarrollo de los ordenadores existen varias simplificaciones. Una de ellas consiste en aprovechar la información del ángulo del gradiente para simplificar la búsqueda ya que entonces basta con sustituir los tres valores  $(x_i, y_i, \theta_i)$  en la ecuación para obtener  $\rho$ .

$$x_i \cos \theta_i + y_i \sin \theta_i = \rho$$

Otra variación es darle un peso a cada píxel dependiendo de su respuesta al gradiente.

La gran ventaja de la transformada de Hough es que utiliza toda la información de la imagen por lo que es más inmune a pérdidas parciales de los bordes que los métodos locales. Sus inconvenientes son, por un lado el coste computacional ya mencionado que es grande, y sobretodo que las rectas son infinitas por lo que no se sabe donde empieza y acaba el segmento de recta presente en la imagen y se puede tomar varios puntos como perteneciente al mismo objeto. Se puede por ejemplo intentar detectar el número y posición de objetos triangulares presentes en la figura 7.6 a partir de la información dada por la transformada de Hough.

#### *Transformada de Hough para circunferencias.*

Al igual que una recta puede definirse parametricamente, se tiene que una circunferencia cumple la curva:

$$(x - x_0)^2 + (y - y_0)^2 = R^2$$

A diferencia del caso anterior ahora se tienen tres parámetros,  $(x_0, y_0, R)$ .

Para un punto perteneciente al borde  $(x_1, y_1)$  el espacio de posibles circunferencia vendrá dado por un cono (figura 7.7-a):

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 - R^2 = 0$$

variando R desde 0 a infinito. Para el punto  $(x_2, y_2)$  se tendrá otro cono (figura 7.7-b) cuya intersección con el primero dará las posibles circunferencias que contienen a ambos puntos (figura 7.7-c). De nuevo se calcularían todas las posibles circunferencias y se elegiría el valor más votado. Al tratarse ahora de un contorno cerrado no se tienen el inconveniente de las rectas infinitas del caso anterior, sin embargo el coste computacional es bastante mayor. Un ejemplo puede verse en la figura 7.8.

#### *Transformada de Hough generalizada.*

La transformada de Hough generalizada se desarrolló para detectar formas que no pudieran expresarse de forma analítica (figura 7.9). Se quiere seguir el esquema anterior en el que cada punto del borde votaba unos valores, eligiéndose después el más votado. Para ello el primer paso consiste en obtener una tabla que represente la forma que se quiere buscar. Para ello:

- Se elige cualquier punto  $(x_0, y_0)$  no perteneciente al borde que se tomará como origen de la forma externa.
- Para cada punto del borde se obtiene la dirección del gradiente,  $\theta$ .
- Se obtiene el módulo y dirección  $(r, \alpha)$  del vector que une  $(x_0, y_0)$  con cada punto del borde.
- Se construye la tabla donde a cada valor del gradiente  $\theta$  se asocian las parejas  $(r, \alpha)$ .

Con ello se cumple que si:

$$\arg(x, y) = \theta$$

entonces:

$$x_0 = x + r \cos(\alpha)$$

$$y_0 = y + r \sin(\alpha)$$

Una vez tomada la imagen en la que se quiere encontrar el objeto se obtendrían los puntos de los bordes; para cada uno de ellos se vería su dirección  $\theta$ , y se iría a la tabla construida inicialmente para ver que valores de  $r$  y  $\alpha$  que hay que aplicar. Con ello ese punto votaría por un origen  $(x_0, y_0)$  determinado. Una vez realizado esto para todos los puntos se elegiría el valor más votado.

Tal como está la tabla solo permite traslaciones respecto de la forma prevista. Para permitir cambios en la escala se deben multiplicar los módulos,  $r$ , por un factor de escala,  $s$ , y para rotaciones se debe añadir un ángulo  $\varphi$ . Se tienen por tanto que buscar cuatro valores  $(x_0, y_0, s, \varphi)$ . Las nuevas fórmulas serían:

$$\arg(x, y) - \varphi = \theta$$

$$x_0 = x + sr \cos(\alpha - \varphi)$$

$$y_0 = y + sr \sin(\alpha - \varphi)$$

## 7.2. Segmentación de regiones.

Como se comentó al principio los objetos presentan una uniformidad en alguna o algunas características. Se buscan por tanto aquellos pixeles que estando unidos presentan una propiedad común.

### 7.2.1. Umbralización

Con la umbralización se convierte una imagen con varios niveles de gris a una nueva con solo dos, de manera que los objetos quedan separados del fondo. La umbralización se basa en que los pixeles de un determinado objeto tienen el mismo nivel de gris. Como esto no será del todo cierto se buscan intervalos de gris, cada uno de ellos pertenecientes a un objeto. Esta técnica será útil si los objetos tienen una superficie parecida y el fondo es uniforme. Un caso típico es un texto, donde las letras destacan del entorno que es blanco o una imagen tomada con iluminación posterior. Otros ejemplos pueden ser imágenes infrarrojas donde los objetos destacan del entorno por su calor. Se aplicarán por tanto las fórmulas:

$$g(x, y) = \begin{cases} 1 & \text{si } T \leq f(x, y) \\ 0 & \text{en cualquier otro caso} \end{cases}$$

cuando el valor buscado sea mayor de uno dado,

$$g(x, y) = \begin{cases} 1 & \text{si } T \geq f(x, y) \\ 0 & \text{en cualquier otro caso} \end{cases}$$

si es menor que el umbral, o

$$g(x, y) = \begin{cases} 1 & \text{si } T_a \leq f(x, y) \leq T_b \\ 0 & \text{en cualquier otro caso} \end{cases}$$

si se conoce que los objetos tienen un cierto intervalo.

El problema está en encontrar los valores de gris a tomar como umbrales entre objetos ya que debido al ruido y a que el objeto y el fondo no tienen un único valor de

gris sino un intervalo, se solapan en algunos valores. Así, en la figura 7.10 aunque se distingue bien el texto de la imagen (a) si se observa su histograma (b) surge la duda de cual es el valor adecuado. Así se observa si se toma un valor alto (c) puntos del fondo se han tomado por letras; lo contrario ocurre si se toma un valor demasiado bajo (e), por último se tiene el mejor umbral (d).

Aunque a continuación se explicarán varias técnicas para obtener los valores para la realización de la umbralización no hay que olvidar que es un problema que depende mucho del caso concreto que se quiera analizar y que la experiencia y conocimiento sobre él determinan en gran manera los valores que se apliquen.

#### *Conocimiento a priori del tamaño del objeto (P-Tile).*

Se tiene un conocimiento a priori de la imagen. Por ejemplo en un texto las letras constituyen el 25% de la imagen. Observando el histograma se toma el valor que delimite ese tanto por ciento. También se puede saber a priori el número de objetos, los valores de gris que pueden presentar, su tamaño, etc.

#### *Mínimo entre máximos.*

Si la imagen sólo constara del objeto y del fondo, el histograma en el caso ideal tendría dos únicos valores de gris. Como en el caso real siempre se va a tener ruido se tendrá algo más parecido a picos. Puede suponerse por tanto que el valor del umbral es el valor mínimo entre esos dos máximos. (figura 7.11). Se debe buscar en el histograma por tanto el máximo valor y el mínimo de los máximos. Para evitar problemas debidos a las variaciones en los histogramas se obtiene primero el máximo, y el nivel de gris que tiene un valor máximo para el producto de su histograma por la distancia al primer máximo. Se busca después el valor mínimo entre ambos valores. En la figura 7.12 se tienen cuatro imágenes donde se han variado la media y desviación típica de los niveles de gris del fondo y de los objetos. Va a servir de comparación entre los métodos que se están explicando.

Existen variaciones sobre el método básico; así se analiza a la vez la imagen del gradiente y se toman de la imagen original solamente los pixeles que tienen un gradiente bajo. Se aumenta así la relación entre los picos y los valles del histograma. Otro enfoque es justo el contrario, se toman los pixeles cuya respuesta la gradiente es alta, suponiendo que serán los pixeles de los bordes, por tanto sus niveles de gris son los que delimitan los objetos. Se analiza el histograma y se toman los valores altos como los umbrales a aplicar.

Se puede suponer que si la imagen fuese perfecta solamente se tendrían dos valores, uno para el fondo y otro para el objeto. Como siempre existirá el ruido gaussiano el histograma sería la suma de las dos gaussianas (figura 7.13). El valor óptimo del umbral sería entonces el corte de las dos gaussianas. Hay que hacer notar que el corte de las dos gaussianas no corresponde al mínimo entre máximos. También habrá casos en los que no se pueda hablar propiamente de la existencia de dos máximos en el histograma. El problema esta, sin embargo, en que la información de que se dispone es el histograma no las gaussianas. Los métodos que se explicarán a continuación hacen suposiciones sobre esas gaussianas y encuentran entonces el valor del umbral.

#### *Medidas iterativas.*



Se supone que el histograma puede asemejarse a dos gaussianas con igual desviación típica. Entonces el punto de cruce debe de estar en el valor intermedio de las dos medias. Como se desconoce su valor se hará una suposición inicial y por un método iterativo nos iremos acercando al valor real. Para ello se supone inicialmente que la media del objeto es el valor más alto y el del fondo el más bajo (o viceversa), se toma por tanto el valor de cruce como el nivel de gris medio de la imagen. A partir de este valor se divide el histograma en dos partes de las que se obtienen sus medias. Se elige el nuevo valor según:

$$T_{k+1} = \frac{1}{2} \left( \frac{\sum_{i=0}^{T_k} i h(i)}{\sum_{i=0}^{T_k} h(i)} + \frac{\sum_{j=1+T_k}^N j h(j)}{\sum_{j=1+T_k}^N h(j)} \right)$$

Se vuelven a obtener las dos medias. Si son distintas se vuelve a obtener un nuevo valor para la umbralización, si no, se para (figura 7.14).

*Método de Otsu para la determinación de umbral.*

De nuevo se supone que el histograma es la suma de dos gaussianas aunque ahora las desviaciones típicas pueden ser distintas. El umbral debe minimizar la suma ponderada de cada una de las varianzas de los objetos presentes, ya que se supone que conforme las gaussianas se aproximen más al histograma real las desviaciones serán menores.

Para determinar los coeficientes se toman las probabilidades de cada uno de los objetos. Así si el umbral es  $t$  se tiene que las probabilidades son:

$$q_1(t) = \sum_{i=1}^t P(i)$$

$$q_2(t) = \sum_{i=t+1}^I P(i)$$

Las medias:

$$\mu_1(t) = \sum_{i=1}^t i \frac{P(i)}{q_1(t)}$$

$$\mu_2(t) = \sum_{i=t+1}^I i \frac{P(i)}{q_2(t)}$$

y las varianzas:

$$\sigma_1^2 = \sum (i - \mu_1(t))^2 \frac{P(i)}{q_1(i)}$$

$$\sigma_2^2 = \sum (i - \mu_2(t))^2 \frac{P(i)}{q_2(i)}$$

Siendo entonces la varianza ponderada

$$\sigma_w^2 = q_1(t) \sigma_1^2 + q_2(t) \sigma_2^2$$

Se recorre todo el rango de los niveles de gris calculándose el valor anterior y se elige aquel umbral que minimiza esta varianza.

Esto puede realizarse de manera más rápida. Todo el histograma tiene una media y desviación típica dada por:

$$\mu = \sum_{i=1}^I iP(i) \quad \sigma^2 = \sum (i - \mu)^2 P(i)$$

La varianza puede reescribirse:

$$\sigma^2 = \sum (i - \mu + \mu_1(t) - \mu_1(t))^2 P(i) + \sum (i - \mu + \mu_2(t) - \mu_2(t))^2 P(i)$$

Si se descompone:

$$\sum (i - \mu_1(t))(\mu_1(t) - \mu)P(i) = \sum i\mu_1(t)P(i) - \sum i\mu P(i) - \sum \mu_1(t)\mu_1(t)P(i) + \sum \mu_1(t)\mu P(i)$$

y como

$$\sum i\mu P(i) = \mu \sum iP(i) = \mu\mu q(t)$$

$$\sum i\mu P(i) = \mu \sum iP(i) = \mu\mu q(t)$$

$$\sum \mu_1(t)\mu_1(t)P(i) = \mu_1(t)\mu_1(t) \sum P(i) = \mu_1(t)\mu_1(t)q(t)$$

$$\sum \mu_1(t)\mu P(i) = \mu_1(t)\mu(t) \sum P(i) = \mu_1(t)\mu q(t)$$

es cero entonces:

$$\sigma^2 = \sum (i - \mu_1(t))^2 + (\mu - \mu_1(t))^2 P(i) + \sum (i - \mu_2(t))^2 + (\mu - \mu_2(t))^2 P(i)$$

$$\sigma^2 = q_1(t)\sigma_1^2 + q_2(t)\sigma_2^2 + q_1(t)(\mu_1(t) - \mu)^2 + q_2(t)(\mu_2(t) - \mu)^2$$

como por otro lado:

$$\begin{aligned} \mu &= q_1(t)\mu_1 + q_2(t)\mu_2 \\ 1 &= q_1(t) + q_2(t) \end{aligned}$$

se tiene que:

$$\sigma^2 = q_1(t)\sigma_1^2 + q_2(t)\sigma_2^2 + q_1(t)(1 - q_1(t))(\mu_1(t) - \mu_2(t))^2$$

por lo que el valor de t que maximice el segundo término minimizará el primero. Para ello ayuda el saber que:

$$\begin{aligned} q_1(t+1) &= q_1(t) + P(t+1) \\ \mu_1(t+1) &= \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)} \end{aligned}$$

y:

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$

El resultado de aplicar este método a las imágenes de la figura 7.12 se puede ver en la figura 7.14.

### *Método de Kittle-Illingworth*

Se vuelve a suponer que se tiene dos gaussianas de medias  $\mu_1$   $\mu_2$  y desviaciones  $\sigma_1$   $\sigma_2$  con una proporciones  $q_1$  y  $q_2$ . Se quiere minimizar:

$$J = \sum_{i=1}^I P(i) \log \frac{P(i)}{f(i)}$$

siendo

$$f(i) = \frac{q_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}\left(\frac{i-\mu_1}{\sigma_1}\right)^2} + \frac{q_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{1}{2}\left(\frac{i-\mu_2}{\sigma_2}\right)^2}$$

¿Por qué se quiere minimizar este valor? Se supone que  $P(i)$  representa la probabilidad de que aparezca un valor de nivel de gris en el histograma de la imagen y, a su vez  $f(i)$  representa la probabilidad de que aparezca un valor de nivel de gris si se supone que la imagen es la suma de las dos gaussianas. Al primer caso lo llamaremos  $X$  y  $Z$  al segundo. Entonces:

$$\text{Prob}(i|X)=P(i)$$

$$\text{Prob}(i|Z)=f(i)$$

Ahora bien  $\text{Prob}(X|i)$  se puede expresar según la fórmula de Bayes:

$$\text{Prob}(X|i) = \frac{P(i) \text{Prob}(X)}{P(i) \text{Prob}(X) + f(i) \text{Prob}(Z)}$$

y de forma análoga:

$$\text{Prob}(Z|i) = \frac{f(i) \text{Prob}(Z)}{P(i) \text{Prob}(X) + f(i) \text{Prob}(Z)}$$

dividiendo:

$$\frac{\text{Prob}(X|i)}{\text{Prob}(Z|i)} = \frac{P(i) \text{Prob}(X)}{f(i) \text{Prob}(Z)}$$

luego:

$$\frac{P(i)}{f(i)} = \frac{\text{Prob}(X|i)}{\text{Prob}(Z|i)} \frac{\text{Prob}(Z)}{\text{Prob}(X)}$$

y ya por fin:

$$\log \frac{P(i)}{f(i)} = \log \frac{\text{Prob}(X|i)}{\text{Prob}(Z|i)} - \log \frac{\text{Prob}(X)}{\text{Prob}(Z)}$$

Luego  $J$  representa el coste de equivocarse al tomar la suma de gaussianas,  $f$ , en lugar del histograma,  $P$ , y por tanto hay que encontrar el valor que lo minimiza.  $J$  puede reescribirse como:

$$J = \sum_{i=1}^I P(i) \log \frac{P(i)}{f(i)} = \sum_{i=1}^I P(i) \log P(i) - \sum_{i=1}^I P(i) \log f(i)$$

al ser el primer término constante, solo hay que maximizar el segundo.

Si las dos distribuciones están bien separadas (lo que no va a ser cierto)

$$\sum_{i=1}^I f(i) = \sum_{i=1}^T \frac{q_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}\left(\frac{i-\mu_1}{\sigma_1}\right)^2} + \sum_{i=T+1}^I \frac{q_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{1}{2}\left(\frac{i-\mu_2}{\sigma_2}\right)^2}$$

y

$$\begin{aligned} \sum_{i=1}^I P(i) \log f(i) &= \sum_{i=1}^T P(i) \log \frac{q_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}\left(\frac{i-\mu_1}{\sigma_1}\right)^2} \\ &\quad + \sum_{i=T+1}^I P(i) \log \frac{q_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{1}{2}\left(\frac{i-\mu_2}{\sigma_2}\right)^2} \end{aligned}$$

$$\sum_{i=1}^I P(i) \log f(i) = \sum_{i=1}^T P(i) \left( \log q_1 - \frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma_1 - \frac{1}{2} \left( \frac{i - \mu_1}{\sigma_1} \right)^2 \right) + \sum_{i=T+1}^I P(i) \left( \log q_2 - \frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma_{21} - \frac{1}{2} \left( \frac{i - \mu_{21}}{\sigma_2} \right)^2 \right)$$

que reordenando:

$$\sum_{i=1}^I P(i) \log f(i) = q_1 \log q_1 + q_2 \log q_2 - \log 2\pi - \frac{1}{2} q_1 \log \sigma_1^2 - \frac{1}{2} \log \sigma_2^2 - \sum_{i=1}^T P(i) \frac{1}{2} \left( \frac{i - \mu_1}{\sigma_1} \right)^2 - \sum_{i=T+1}^I P(i) \frac{1}{2} \left( \frac{i - \mu_{21}}{\sigma_2} \right)^2$$

siendo los dos últimos sumatorios igual a un medio luego:

$$-\sum_{i=1}^I P(i) \log f(i) = \frac{1 + 2 \log 2\pi}{2} - q_1 \log q_1 - q_2 \log q_2 + \frac{1}{2} (q_1 \log \sigma_1^2 + \log \sigma_2^2)$$

para evaluarlos de forma iterativa falta la formula para la desviación típica que es:

$$\sigma_1^2(t+1) = \frac{q_1(t) \left[ \sigma_1^2(t) + (\mu_1(t) - \mu_1(t+1))^2 \right] + P(t+1) \left( (t+1) - \mu_1(t+1) \right)^2}{q_1(t+1)}$$

$$\sigma_2^2(t+1) = \frac{(1 - q_1(t)) \left[ \sigma_2^2(t) + (\mu_2(t) - \mu_2(t+1))^2 \right] - P(t+1) \left( (t+1) - \mu_2(t+1) \right)^2}{1 - q_1(t+1)}$$

El resultado de aplicar este método a las imágenes de la figura 7.12 se puede ver en la figura 7.14.

*Uso de la entropía. Método de Pum.*

Se define la entropía H como

$$H = -\sum p_i \log(p_i)$$

Se busca entonces maximizar la suma de las dos entropías. Esto puede verse que es igual a maximizar:

$$f(t) = \frac{H_t}{H_T} \frac{\log P_t}{\log(\max(p_0, p_1, \dots, p_t))} + \left[ 1 - \frac{H_t}{H_T} \right] \frac{\log(1 - P_t)}{\log(\max(p_{t+1}, p_{t+2}, \dots, p_N))}$$

Siendo:

$$H_t = -\sum_{i=0}^{i=t} p_i \log(p_i)$$

$$H_T = -\sum_{i=0}^{i=N} p_i \log(p_i)$$

$$P_t = \sum_{i=0}^{i=t} p_i$$

*Uso de la entropía. Método de Kapur.*

Kapur considera que en lugar de definir la probabilidad de que un píxel pertenezca a un nivel de gris como:

$$\frac{p(i)}{P(255)}$$

debería ser:

$$\frac{p(i)}{P_T} \quad \frac{p(i)}{1 - P_T}$$

Las ecuaciones de la entropía tienen entonces la fórmula:

$$H_t = - \sum_{i=0}^{i=t} \frac{p_i}{P_T} \log\left(\frac{p_i}{P_T}\right)$$

$$H_T = - \sum_{i=0}^{i=N} \frac{p_i}{1 - P_T} \log\left(\frac{p_i}{1 - P_T}\right)$$

Se debe entonces encontrar el nivel de gris que maximiza:

$$H_t + H_T$$

Los resultados de estos dos últimos métodos pueden observarse en la figura 7.15.

#### *Umbralización recursiva.*

Hasta ahora los métodos vistos suponen que el histograma de la imagen hay dos niveles de gris dominantes por lo que el histograma es binodal. Puede darse el caso de que sea más parecido al presentado en la imagen 7.16 con tres valores de gris. Para detectar los objetos se aplicaría cualquiera de los métodos vistos antes (mínimo entre máximos por ejemplo). Después de la separación se vuelve a analizar cada uno de los dos histogramas para ver si es necesario aplicar una nueva umbralización o no (por ejemplo, si hay otros dos máximos o no). Esto se haría repetitivamente hasta que no hiciese falta ninguna umbralización más. En la figura 7.17 puede observarse una imagen con tres zonas claramente diferenciadas. Si se obtiene su histograma se verá que presenta tres máximos. Con la umbralización recursiva pueden calcularse tres nuevas imágenes. Se hace notar que los bordes de las letras aparecen en dos imágenes. Ello es debido a un problema de muestreo por lo que habría que realizar un procesamiento posterior para eliminarlas tal como se verá en el capítulo de transformaciones morfológicas.

#### *Umbralización basada en el color.*

Es el mismo caso visto con anterioridad solo que se tendrá que realizar una segmentación para cada una de las componentes del color. Cuando se trabaja en el sistema RGB, el color de cada uno de los puntos de una imagen está definido por tres componentes: las componentes roja, verde y azul. Para identificar los puntos de un determinado color se comprueba que los niveles de sus tres componentes corresponden a los del color buscado. Hay que resaltar que "rojo" no se corresponde con la componente "roja" de la imagen, (ya que el color blanco tiene también una componente roja máxima), sino con aquellos pixeles que tengan máximo de rojo y mínimo de las otras dos componentes. Por ello una umbralización por color tendrá la forma de:

$$g(x,y) = \begin{cases} R_a \leq f_r(x,y) \leq R_b \\ 1 \text{ si } V_a \leq f_v(x,y) \leq V_b \\ A_a \leq f_a(x,y) \leq A_b \\ 0 \text{ en cualquier otro caso} \end{cases}$$

Donde  $f_r(x,y)$ ,  $f_v(x,y)$  y  $f_a(x,y)$  son, respectivamente, las funciones que dan los niveles de rojo, verde y azul de cada uno de los puntos de la imagen. Así en la figura 7.18 se tiene una imagen con sus tres componentes. Se haría la umbralización para cada una de ellas con lo que se tienen los pixeles que cumplen la condición para ese color.

Después habría que realiza una operación lógica AND entre las tres ya que los pixeles que interesan son los que han pasado la umbralización en las tres componentes. Al igual que en el método anterior existen pixeles que no interesan, y que también se eliminarán posteriormente con ayuda de las transformaciones morfológicas.

El uso del espacio RGB presenta un problema: ¿Qué ocurre si la iluminación cambia? La mejor solución es trabajar con el espacio HSI que teóricamente es independiente de la iluminación. Debido al coste computacional de pasar de RGB a HSI existen variaciones de la umbralización en RGB vista antes; así mas que trabajar con unas bandas estáticas se usará relación entre las componentes. Así se tendrá una nueva función de umbralización (suponiendo que se elige la componente roja como referencia):

$$g(x,y) = \begin{cases} 1 & \text{si } \begin{cases} R_a \leq f_r(x,y) \leq R_b \\ V'_a \leq \frac{f_v(x,y)}{f_r(x,y)} \leq V'_b \\ A'_a \leq \frac{f_a(x,y)}{f_r(x,y)} \leq A'_b \end{cases} \\ 0 & \text{en cualquier otro caso} \end{cases}$$

donde se supone que si una componente sube o disminuye por efecto de la iluminación las otras lo harán en la misma proporción por lo que sus cocientes serán constantes.

#### *Histogramas laterales.*

Otra posibilidad es la utilización de los histogramas laterales, en los que se suman los valores de los pixeles por columnas y por filas. Se aplicarán cualquiera de las técnicas antes vistas para los histogramas del valor de gris y se delimita la imagen. Serán útiles para determinar en que zona se encuentran los objetos. En el ejemplo de la Figura 7.19 se trata de localizar donde se haya presente la circunferencia.

#### *Umbralización adaptativa.*

Las imágenes en las que la iluminación no sea constante presentan el problema de que puntos que pertenezcan al mismo objeto presenten distintos valores en sus niveles de gris. Para separar el objeto del fondo se divide la imagen en varias subimágenes de dimensiones nxm donde se supone que los cambios de la iluminación no son fuertes. A cada una de las subimágenes se aplicaría alguno de los métodos vistos con anterioridad con los que el umbral se va adaptando a los cambios de la iluminación.

La limitación de los métodos basados en el histograma reside en que el histograma no tiene en cuenta la información espacial, solo el conjunto de intensidades de toda la imagen. Imágenes bien distintas pueden tener el mismo histograma. Así, solamente con el histograma no se puede distinguir entre una imagen binaria y otra que sea una distribución aleatoria de ceros y unos.

### **7.2.2. Etiquetado.**

Con la umbralización los objetos han sido separados del fondo. Sin embargo al utilizar únicamente el histograma no han sido distinguidos entre sí (todos tienen el mismo nivel de gris alto); es por ello que la siguiente etapa es el etiquetado (labelling en

la literatura anglosajona), o lo que es lo mismo a cada objeto se le asignará un nivel de gris distinto. Para ello, y empezando por el píxel superior izquierdo (figura 7.20) se empieza a recorrer la imagen en dirección a la derecha. Cuando se encuentra el primer píxel que esté a nivel alto se le asocia la etiqueta (*label*) 1, para identificarlo como primer objeto. Se examinan a continuación sus vecinos para ver si también están a nivel alto, si lo están recibirán la misma etiqueta. Cuando un píxel no sea vecino de uno etiquetado pero esté a nivel alto se le asocia la siguiente etiqueta, 2, etc. Con un solo recorrido de la imagen se pueden producir indeterminaciones como puede verse en la figura 7.21. ¿Se le asocia al píxel la etiqueta 1 ó 2? La solución está en llevar una tabla con etiquetas que corresponden al mismo objeto. Una vez terminada esta segunda pasada sobre la imagen se deshacen las ambigüedades asignando a todos los puntos la misma etiqueta (figura 7.22). ¿Qué ocurre con las etiquetas que quedan libres? Aunque no es necesario, los algoritmos de etiquetado asignan esa etiqueta al siguiente objeto de tal forma que no existan etiquetas libres entre el valor mínimo y máximo. Una ventaja de esto es lo fácil que resulta obtener el número de objetos distintos ya que solo hay que obtener el histograma de la imagen etiquetada y ver el nivel de gris más alto que tenga algún píxel en la imagen.

### 7.2.3. Crecimientos de regiones.

El crecimiento de regiones consiste en elegir a una serie de píxeles iniciales, denominados puntos semilla; a los que se les irán añadiendo píxeles de iguales propiedades. Para ello se examina si los píxeles contiguos tienen valores parecidos para la o las características que tienen los puntos semilla. Si es así pertenecen a la misma región y pasan a tener los mismos valores que los puntos iniciales. Es decir si se cumple que la diferencia entre los niveles de gris, por ejemplo, es menor que un valor prefijado:

$$|f(x, y) - f(s_{xi}, s_{yi})| < T$$

pertenecerán al mismo objeto que ese punto semilla. Esto se comprueba para los ocho vecinos para después comparar cada uno de los vecinos de los píxeles que se hayan asignado a alguna región. Es un método iterativo que termina cuando ningún píxel cambia sus propiedades.

Por ejemplo en la imagen siguiente se han elegido dos puntos semillas con niveles de gris uno y siete.

0	1	5	6	5
2	0	7	6	6
0	<b>1</b>	6	7	7
1	1	5	8	7
0	6	7	0	5

Si la diferencia entre dos píxeles se toma menor o igual a tres se obtienen dos regiones:

a	a	b	b	b
a	a	b	b	b
a	<b>a</b>	b	<b>b</b>	b
a	a	b	b	b
a	a	b	b	b

Un problema es cómo determinar los puntos iniciales. Aunque una solución es “lanzar” las semillas al azar resulta más lógico elegir pixeles cuyo nivel de gris corresponda a picos del histograma. Habrá que prever por otro lado que se llegue a un punto muerto donde queden pixeles por clasificar pero las regiones no puedan crecer más, por lo que es necesario volver a lanzar más semillas. El gran inconveniente del crecimiento de regiones es que puede dar diversos resultados dependiendo de donde caigan las semillas y del umbral que se utilice para determinar cuándo un píxel pertenece o no a una región. Así si se toma la figura 7.23 se observa en el caso (a) como el nivel de gris del fondo sigue una rampa de menos a más, mientras que el objeto sigue otra en sentido contrario. En el caso (b) las semillas han caído en el centro del objeto y de cada lado del fondo mientras que el umbral  $T$  es la mitad del salto en los niveles de gris. Con esto la clasificación se realiza de forma correcta. Sin embargo en el caso (c) las semillas del objeto han caído en los dos extremos con lo que el resultado es que la imagen puede separarse en cuatro zonas distintas. Peor resultado se observa en el caso (d) donde ahora el umbral toma la mitad del valor de los casos anteriores con lo que se obtienen siete zonas distintas.

El crecimiento de regiones dará por tanto más objetos de los presentes en la imagen. Así en la figura 7.24 se tiene el resultado de la segmentación de la figura superior. En el caso de la izquierda se ha asignado a cada región un nivel de gris distinto según la semilla y en la imagen inferior derecha se ha sustituido por el nivel de gris medio de la región. Puede observarse como zonas comunes se han partido en regiones más pequeñas.

Para solucionar esto se puede unir el resultado del crecimiento de regiones con un análisis de los bordes detectados en la imagen original. Así se puede saber si la separación entre dos regiones responde a un salto brusco en el gradiente o a un paso suave. Ahora no importa que se pierdan puntos de los bordes debido al ruido ya que tenemos la información obtenida por el crecimiento de regiones.

#### 7.2.4. División y unión de regiones.

La división y unión de regiones (split & merge en la literatura anglosajona) esta basada en la estructura quadtree (figura 7.25) que permite representar objetos (en nuestro caso imágenes) con varias resoluciones. Así, como puede verse en la figura de la derecha, de cada nodo del árbol pueden salir cuatro ramas. Debido al criterio de numeración (figura izquierda) se conoce inmediatamente qué partes de la capa inferior son vecinas de qué nodos de la superior. Viendo los valores numéricos y los nodos es fácil decir que la imagen tiene tres objetos dos con valor alto y otro con nivel bajo. Para llegar a esa representación el método basado en la división y unión de regiones (figura 7.26) consta de dos partes: dividir una imagen en varias para después buscar subimágenes iguales. Se comienza así con toda la imagen. Si toda ella cumple una determinada característica se deja como está. Si no, se divide *-split-* en cuatro subimágenes iguales.

Así se obtendría la media y desviación típica de la imagen:

$$\mu = \frac{1}{n} \sum f(x, y)$$

$$\sigma^2 = \frac{1}{n} \sum (f(x, y) - \mu)^2$$



y si la desviación típica es mayor que un valor, lo que indica gran variación en los niveles de gris de la imagen, se divide. Otro criterio es analizar el histograma buscando varios picos.

Después se volvería a analizar la desviación típica de cada subimagen. Si vuelven a ser superiores al umbral se marcan para dividir las en la siguiente iteración. Si no, no volverán a dividirse y se intenta unirla *-merge-* a alguna región vecina que tampoco vaya a ser dividida. Para ello se comparan las medias de ambas regiones con la desviación típica:

$$|\mu_1 - \mu_2| \leq k\sigma$$

Si no se cumple serán regiones independientes.

La segmentación llega a termino después de un número finito de iteraciones. Así, por ejemplo, una imagen de 512x512 lleva 9 iteraciones ( $2^9 = 512$ ). De todas formas se pone un número máximo de iteraciones ya que si no el coste computacional sería prohibitivo. Es por ellos que este método no preserva los contornos reales de los objetos como puede apreciarse en la figura 7.27.

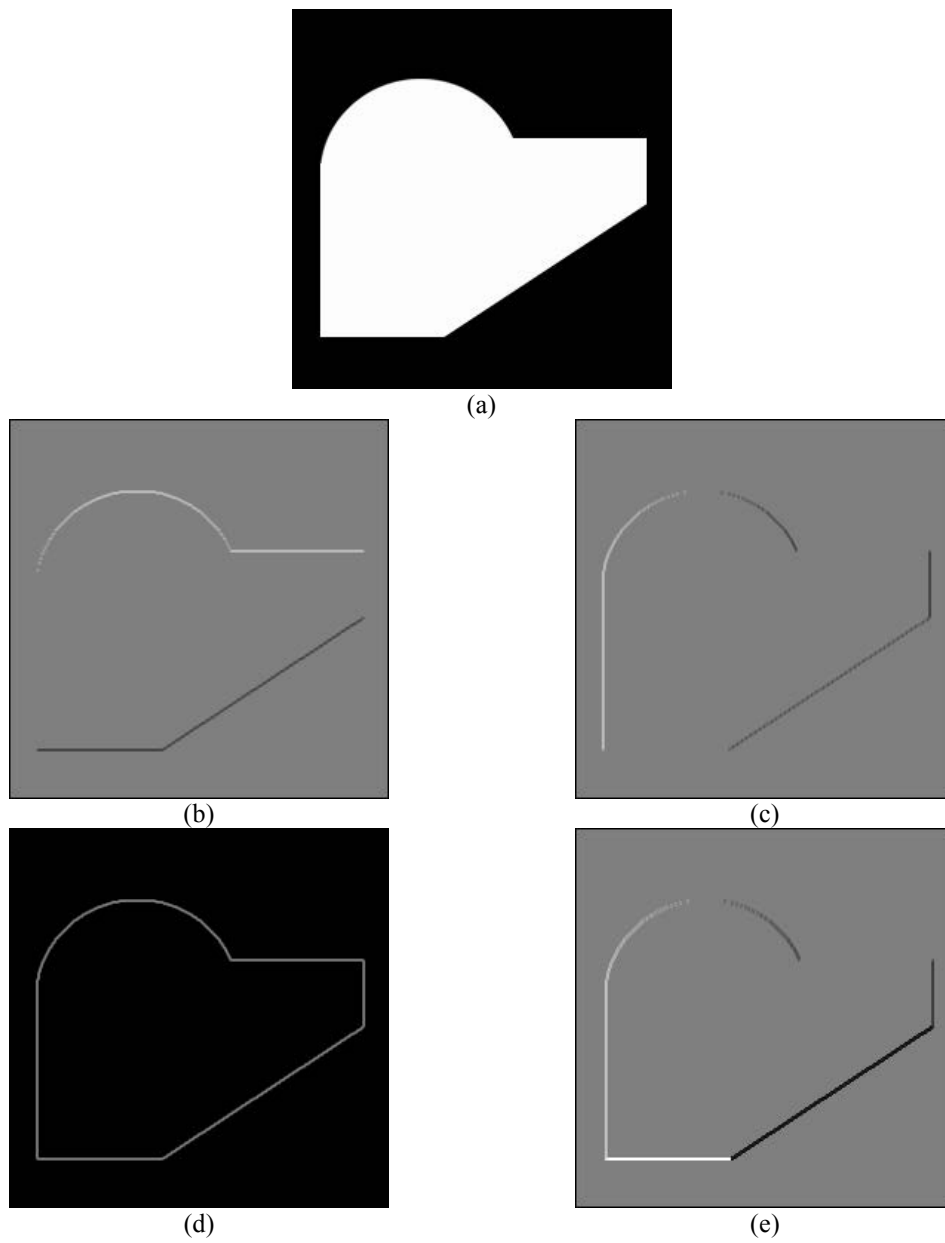


Figura 7.1 Segmentación basada en la detección de bordes. (a) Pieza original (b) gradiente vertical (c) gradiente horizontal (d) módulo del gradiente (e) dirección del gradiente

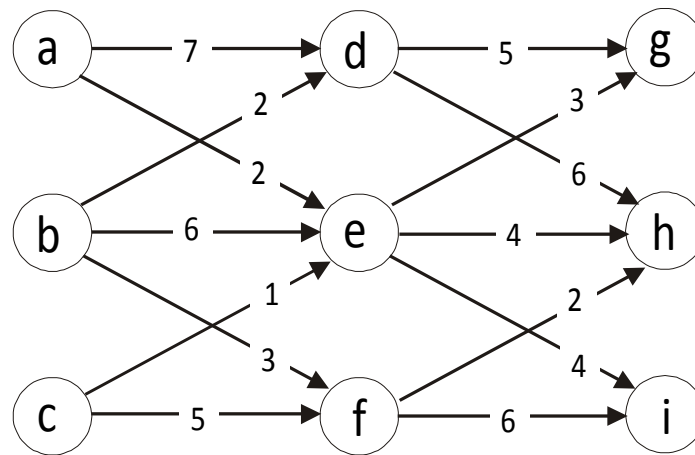


Figura 7.2. Determinación del camino mediante programación dinámica.

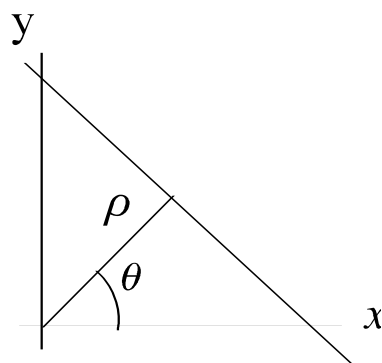


Figura 7.3. Ecuación paramétrica de una recta.

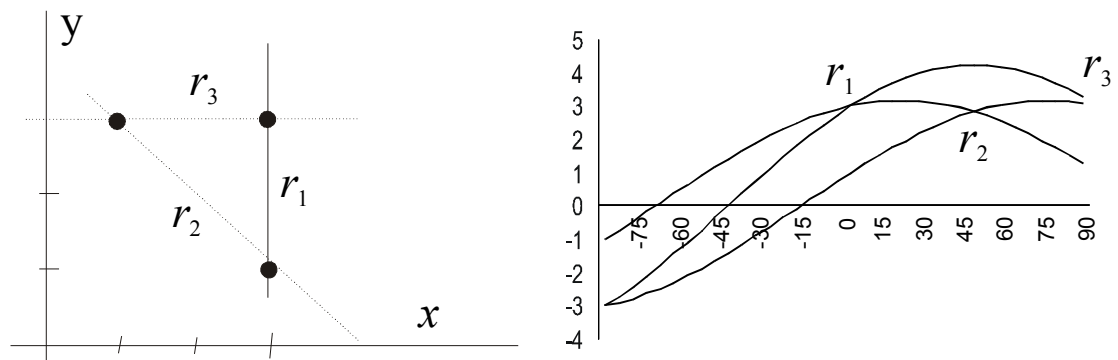


Figura 7.4. La transformada de Hough

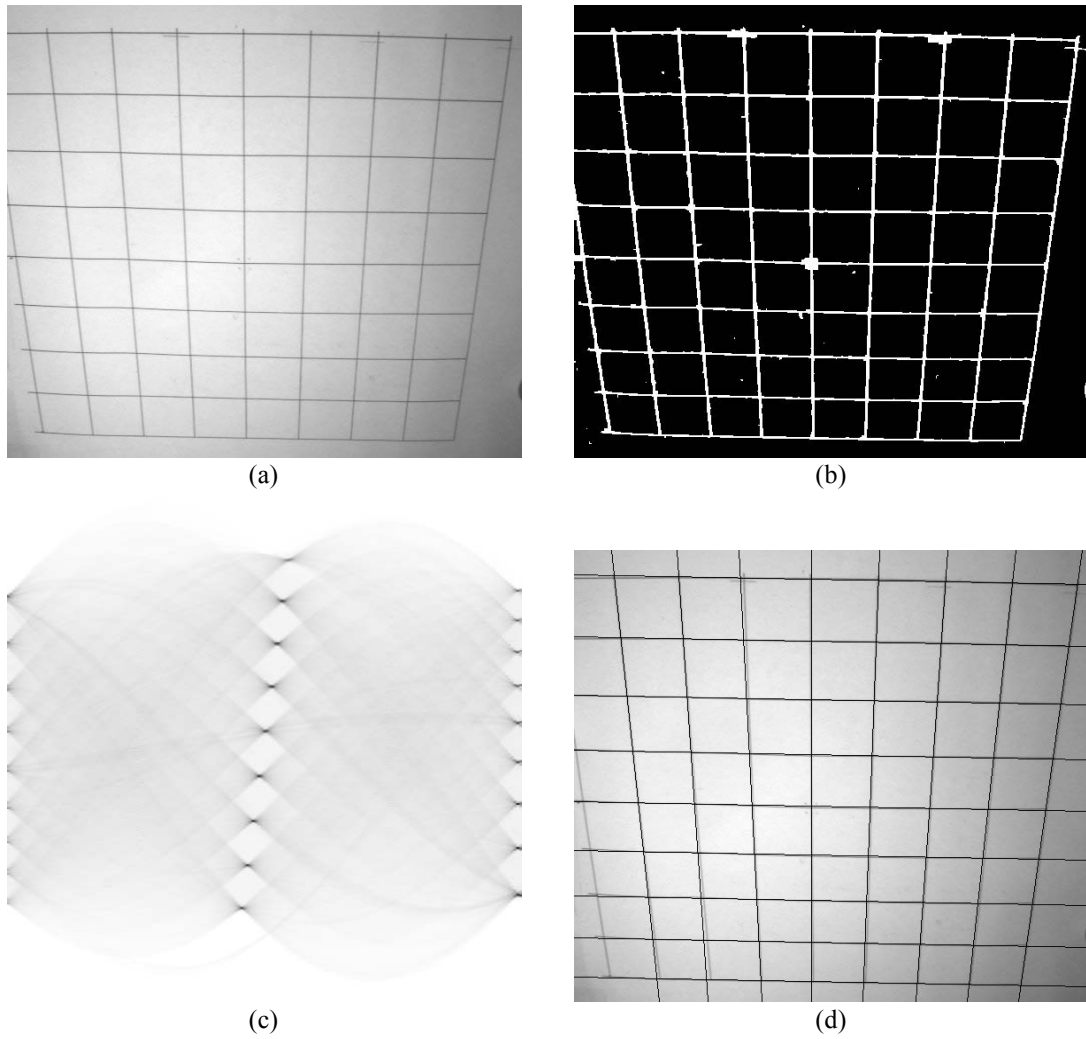


Figura 7.5. La transformada de Hough (a) imagen original (b) gradiente (c) representación de las rectas en el espacio de parámetros  $\theta$  y  $\rho$  (d) rectas detectadas

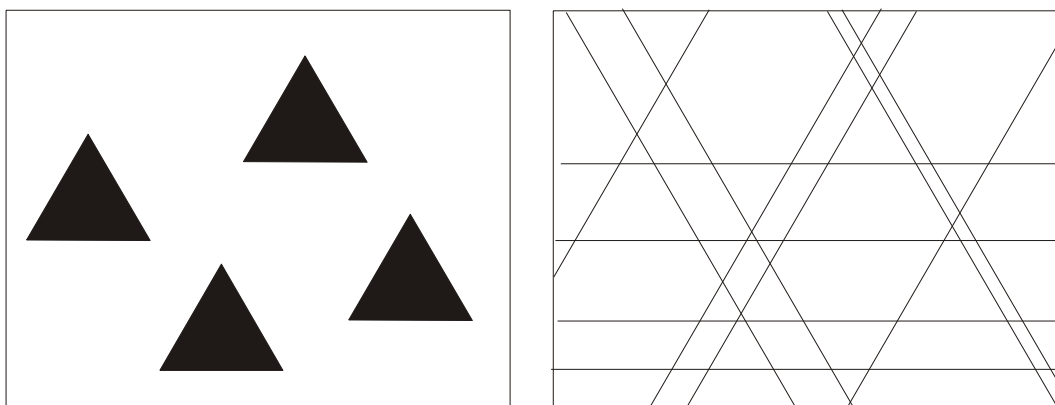


Figura 7.6. Problemas de la transformada de Hough para rectas.

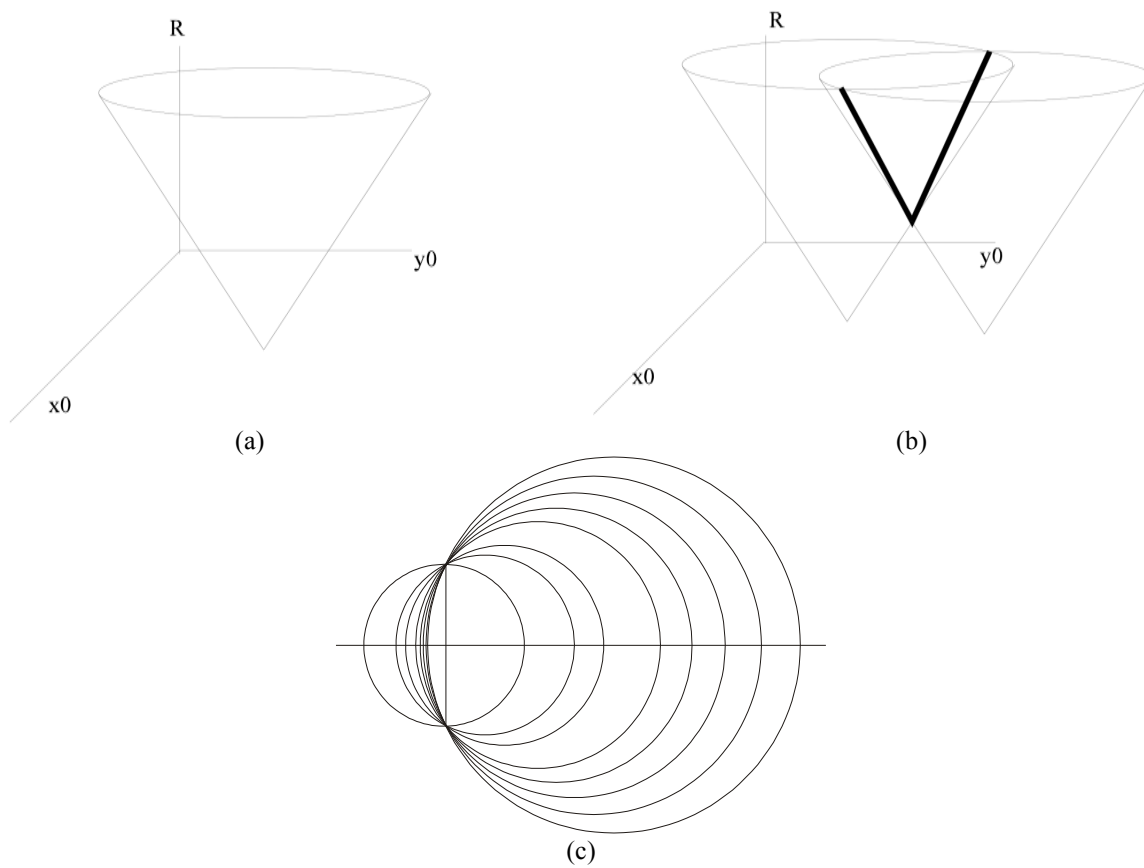


Figura 7.7. Transformada de Hough para circunferencias.

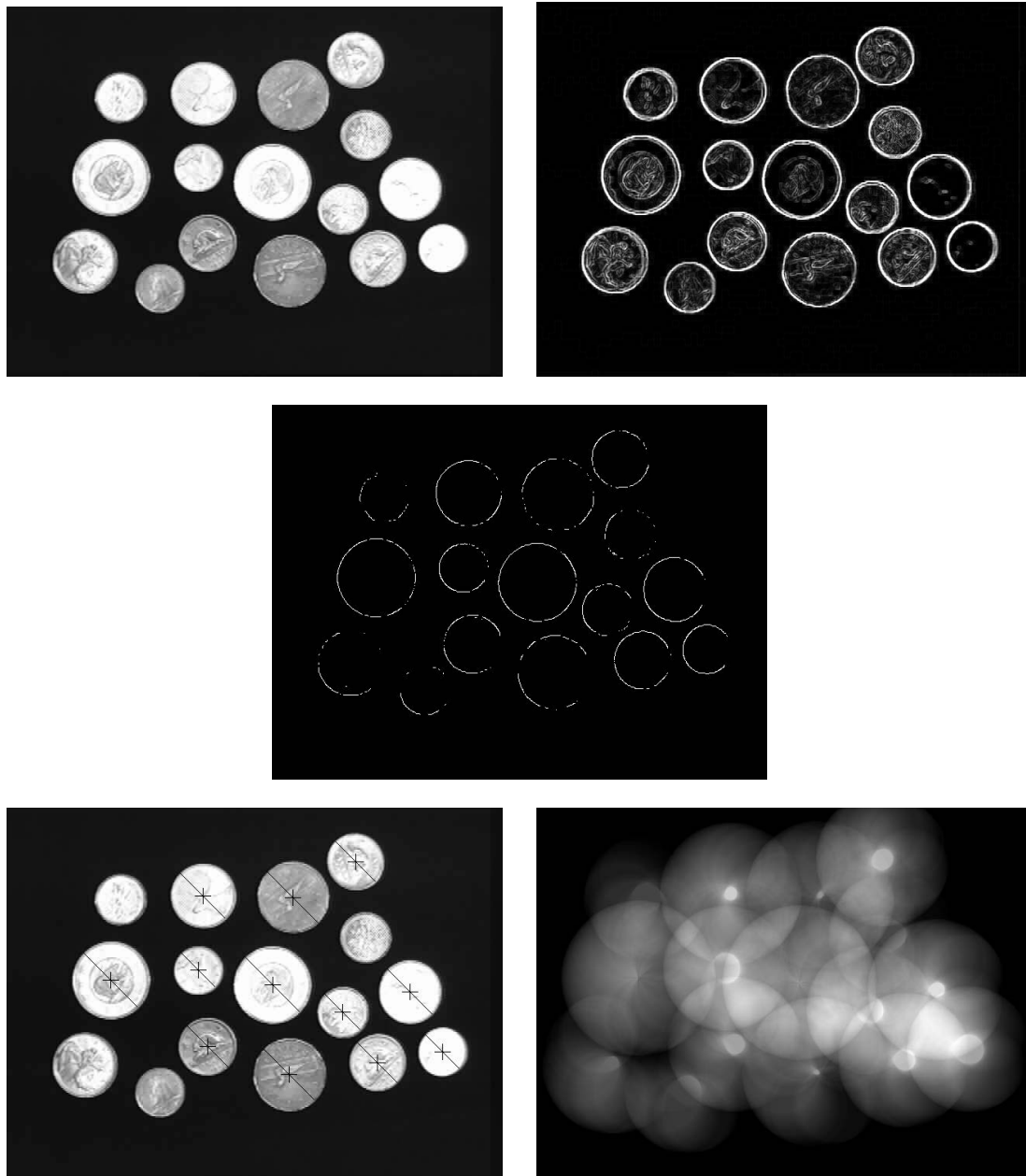


Figura 7.8. Transformada de Hough para circunferencias. (a) imagen original (b) gradiente (c) determinación de los puntos de borde (d) detección de las 12 circunferencias con mayor número de puntos (e) Proyección del espacio XYR sobre el plano XY.

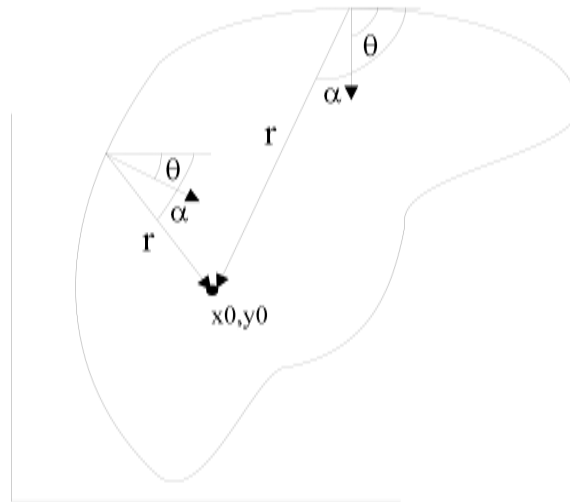


Figura 7.9. Transformada generalizada de Hough

Aunque a continuación se explicarán varias técnicas de la umbralización no hay que olvidar q el caso concreto que se quiera analizar y que la determina en gran manera los valores que se aplique

La decisión puede ser tomada siguiendo:

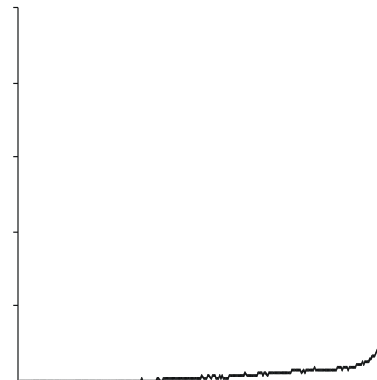
- Se tiene un conocimiento a priori de la imagen constituyen el 25% de la imagen. Observand delimita ese tanto por ciento. También se puede valores de gris que pueden presentar, su tamaño
- Otra posibilidad es buscar un mínimo entre dos a una curva de la forma :

con mínimo en:

$$x = -\frac{b}{2a}$$

- Umbralización recursiva. Después de la separa para ambas imágenes hasta que no exista ningú

(a)



(b)

Aunque a continuación se explicarán varias técnicas de la umbralización no hay que olvidar q el caso concreto que se quiera analizar y que la determina en gran manera los valores que se aplique

La decisión puede ser tomada siguiendo:

- Se tiene un conocimiento a priori de la imagen constituyen el 25% de la imagen. Observand delimita ese tanto por ciento. También se puede valores de gris que pueden presentar, su tamaño
- Otra posibilidad es buscar un mínimo entre dos a una curva de la forma :

con mínimo en:

$$x = -\frac{b}{2a}$$

- Umbralización recursiva. Después de la separa para ambas imágenes hasta que no exista ningú

(c)

Aunque a continuación se explicarán varias técnicas de la umbralización no hay que olvidar q el caso concreto que se quiera analizar y que la determina en gran manera los valores que se aplique

La decisión puede ser tomada siguiendo:

- Se tiene un conocimiento a priori de la imagen constituyen el 25% de la imagen. Observand delimita ese tanto por ciento. También se puede valores de gris que pueden presentar, su tamaño
- Otra posibilidad es buscar un mínimo entre dos a una curva de la forma :

con mínimo en:

$$x = -\frac{b}{2a}$$

- Umbralización recursiva. Después de la separa para ambas imágenes hasta que no exista ningú

(d)

Aunque a continuación se explicarán varias técnicas de la umbralización no hay que olvidar q el caso concreto que se quiera analizar y que la determina en gran manera los valores que se aplique

La decisión puede ser tomada siguiendo:

- Se tiene un conocimiento a priori de la imagen constituyen el 25% de la imagen. Observand delimita ese tanto por ciento. También se puede valores de gris que pueden presentar, su tamaño
- Otra posibilidad es buscar un mínimo entre dos a una curva de la forma :

con mínimo en:

$$x = -\frac{b}{2a}$$

- Umbralización recursiva. Después de la separa para ambas imágenes hasta que no exista ningú

(e)

Figura 7.10. Efecto de la umbralización.(a) imagen original (b) histograma (c) umbral alto (d) umbral intermedio (e) umbral bajo

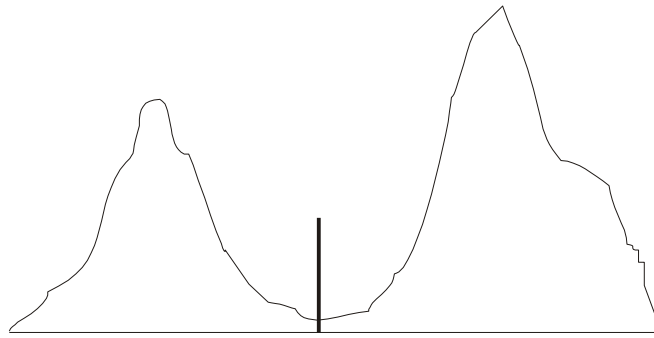


Figura 7.11. Mínimo entre máximos.

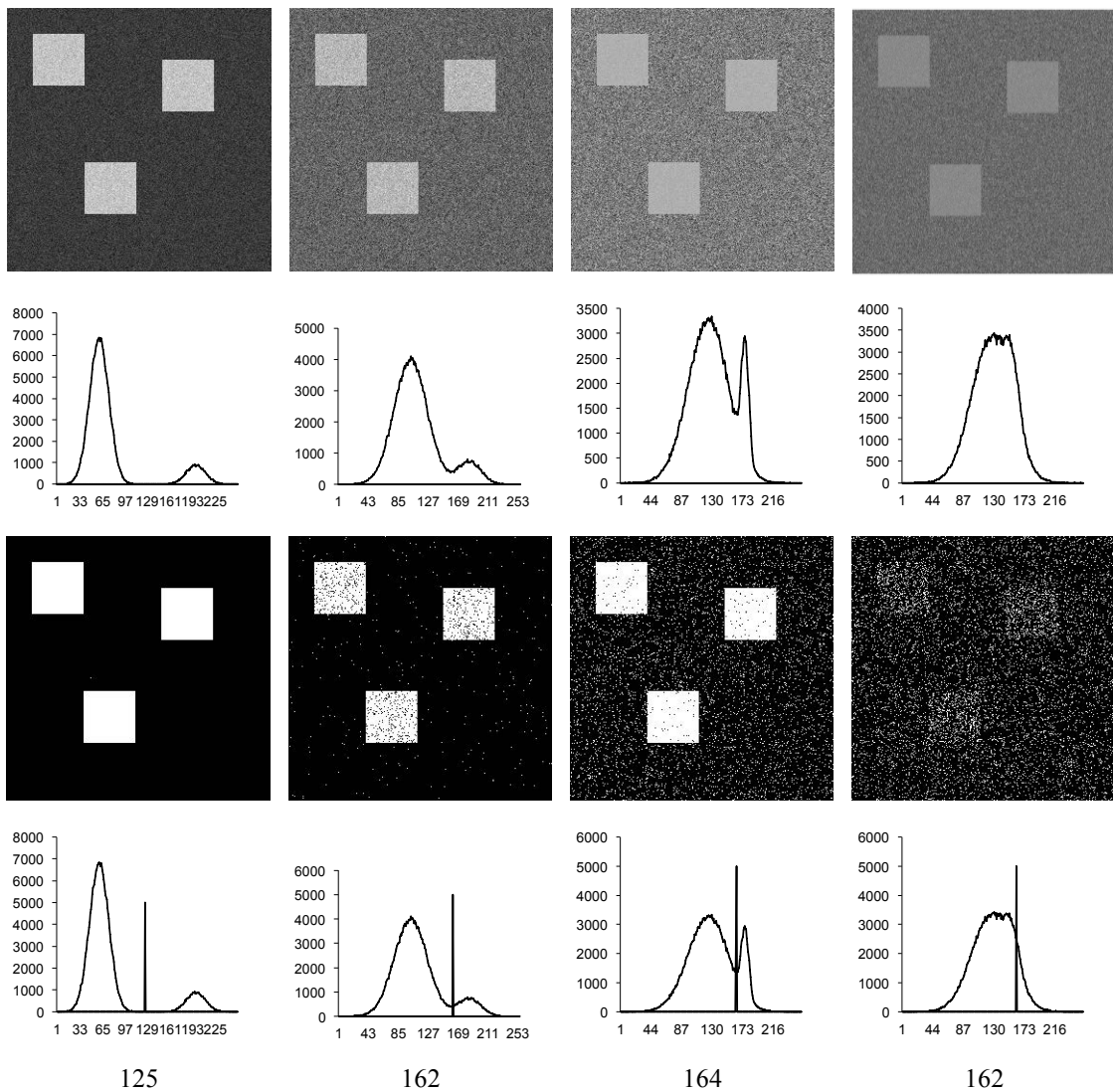


Figura 7.12 Imágenes originales y mínimo entre máximos.



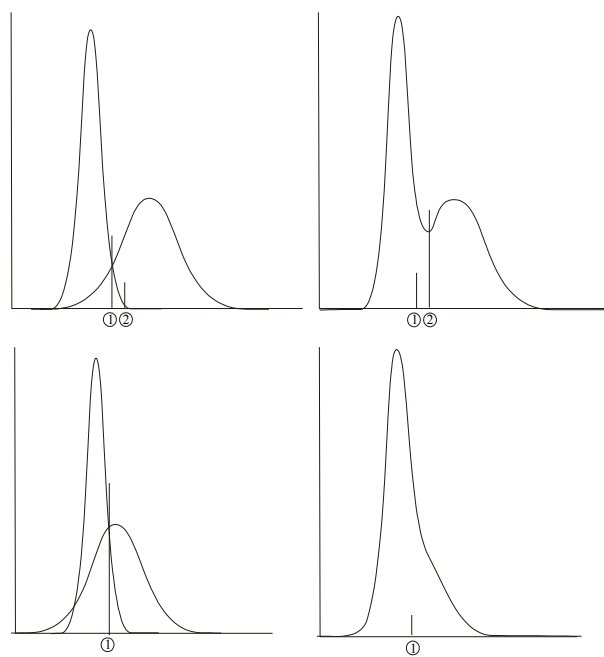


Figura 7.13. Comparación entre mínimo entre máximos y cruce de gaussianas.

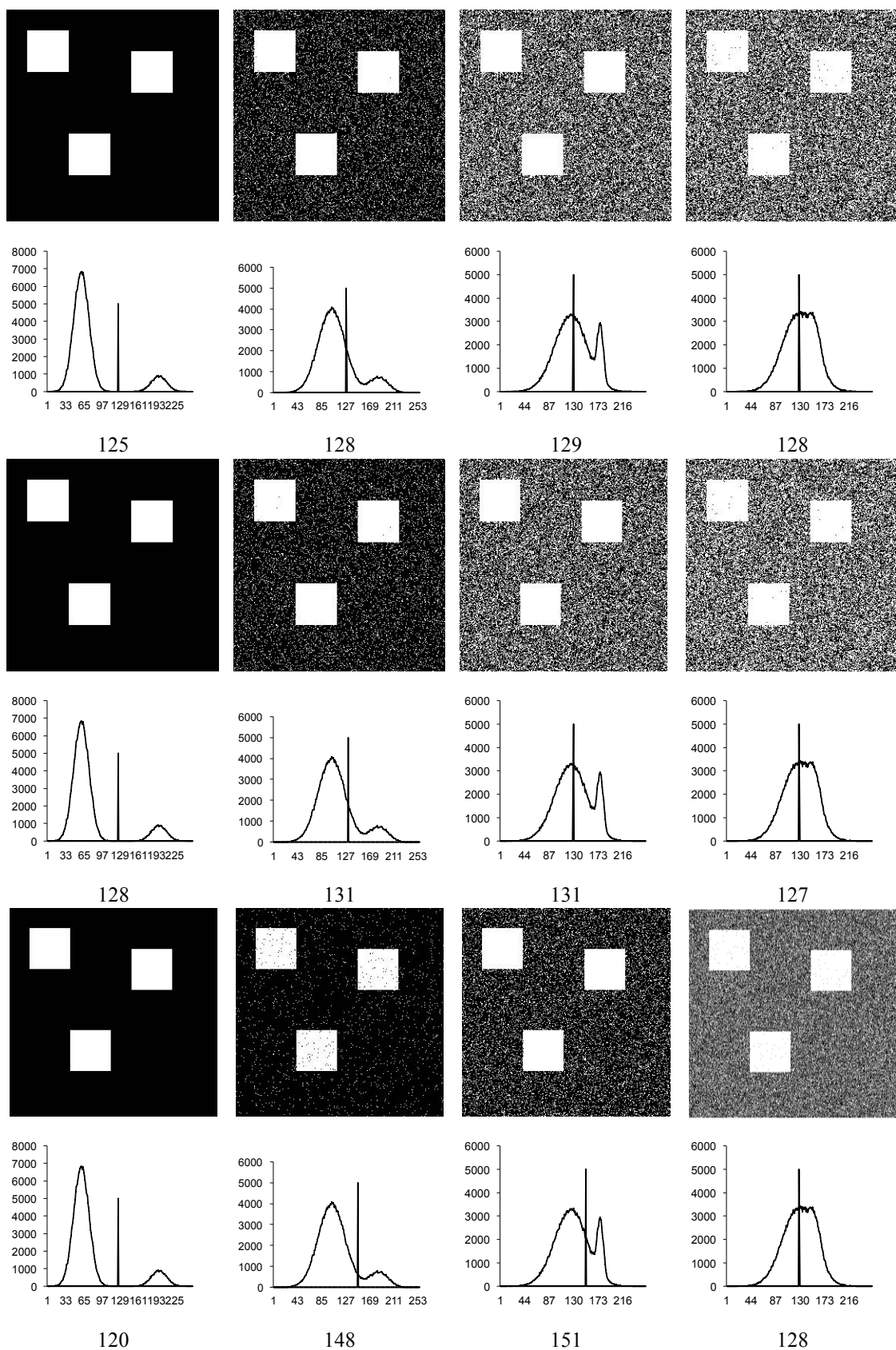


Figura 7.14 Medidas recursivas, Otsu y Kittler-Illingworth.

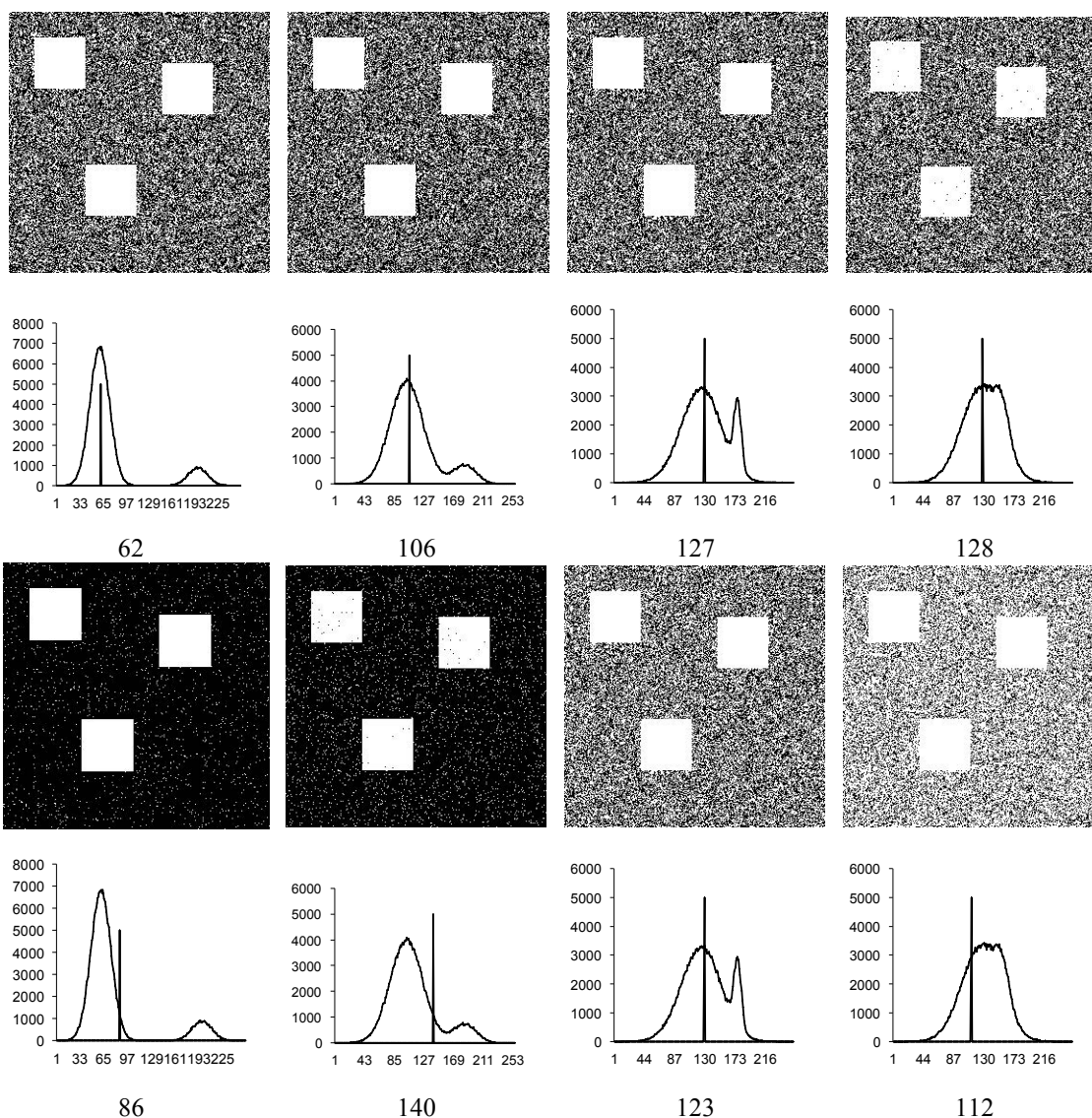


Figura 7.15 Pun y Kapur

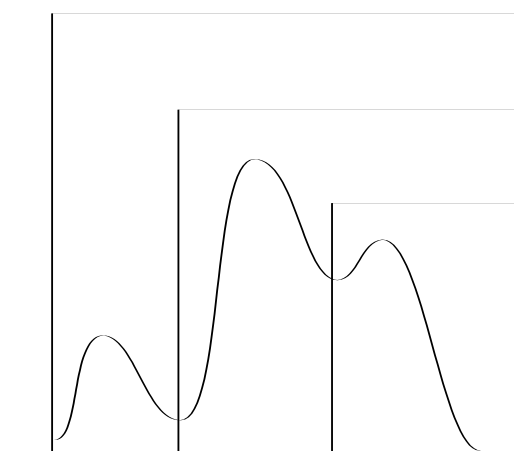


Figura 7.16. Umbralización recursiva.

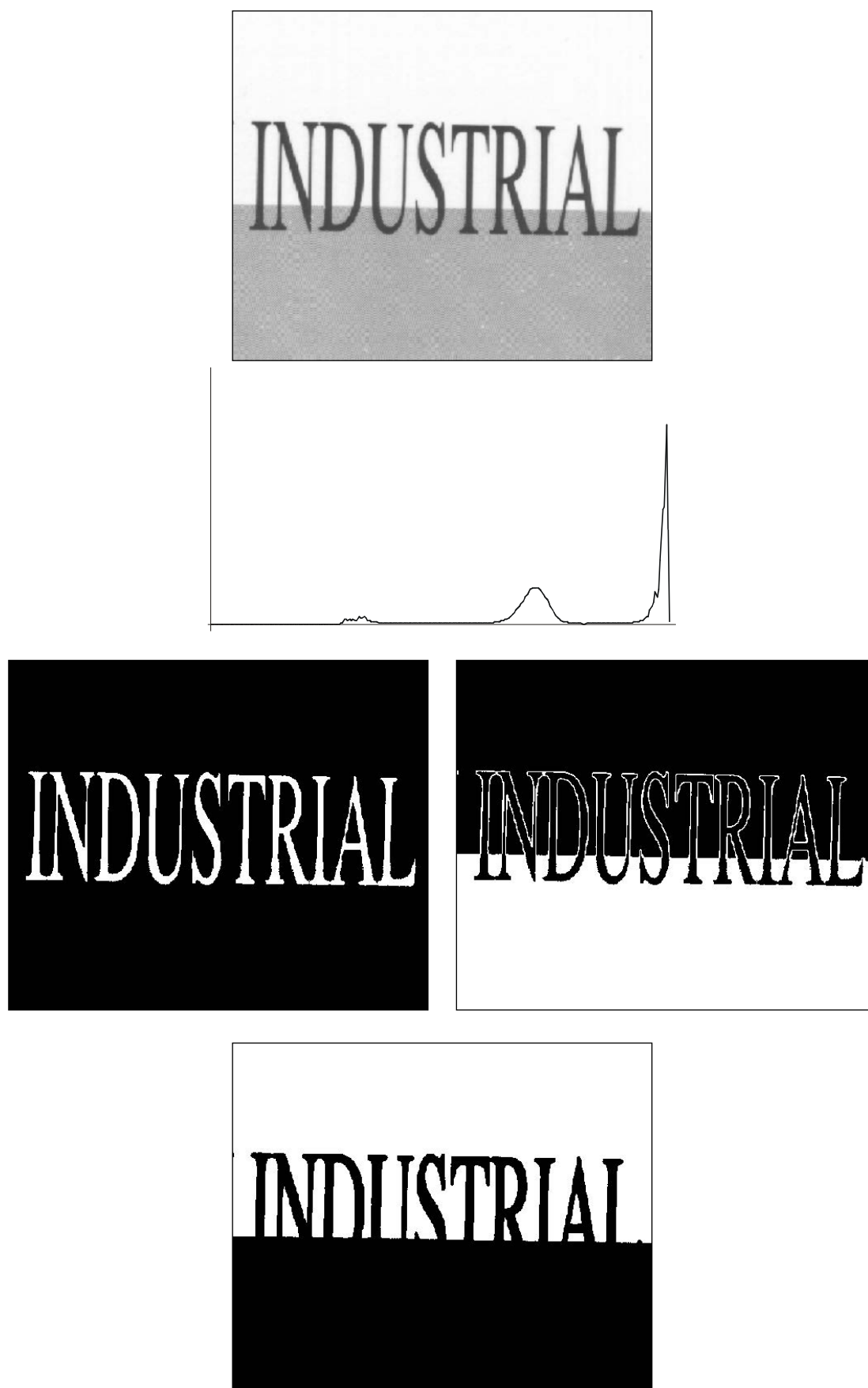


Figura 7.17. Umbralización recursiva.



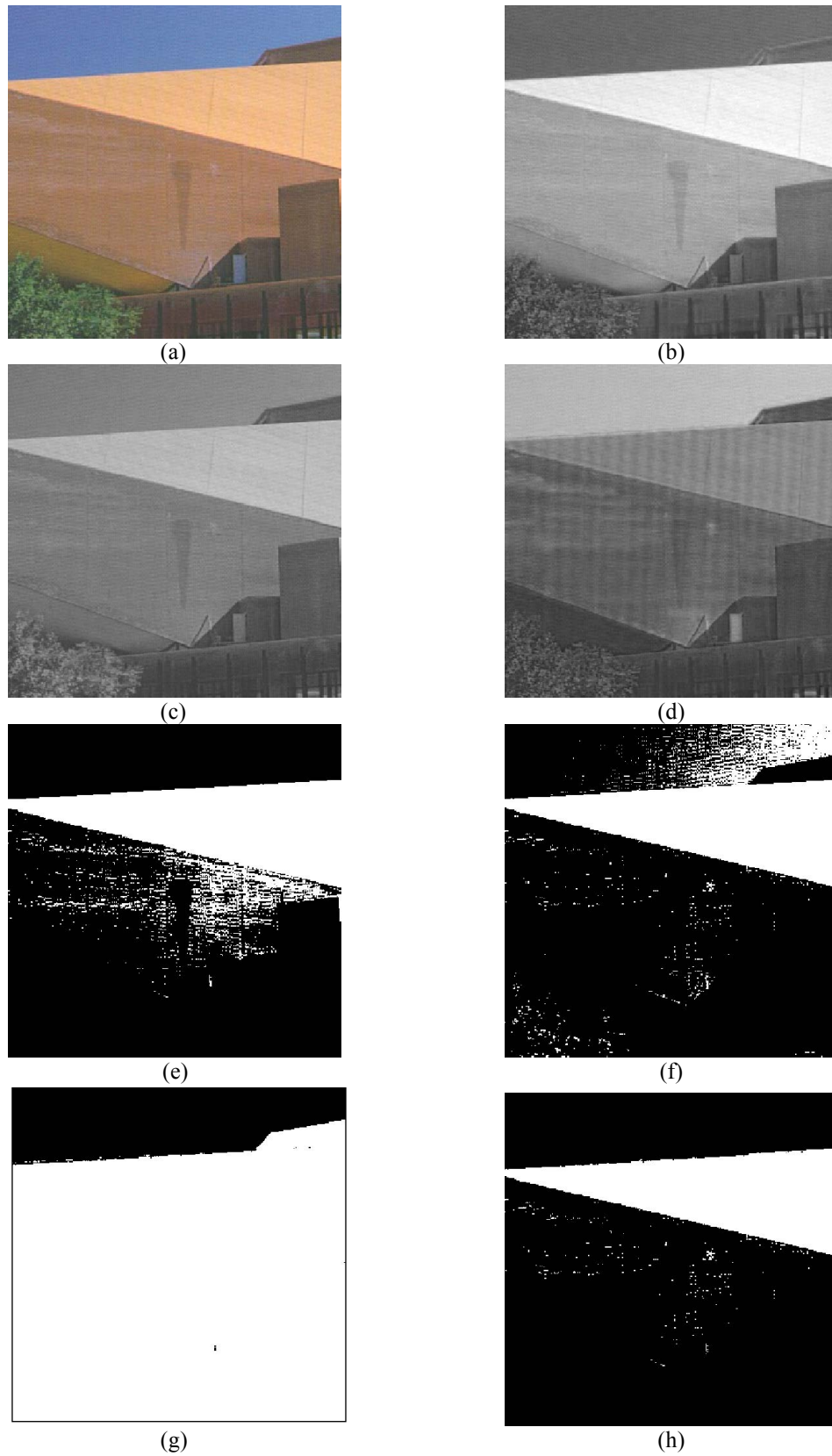


Figura 7.18. Umbralización por color. (a) Imagen original (b), (c) y (d) componentes roja, verde y azul (e) (f) y (g) resultado de umbralizar cada componente por su parte (h) resultado de aplicar la operación lógica AND para las tres umbralizaciones.

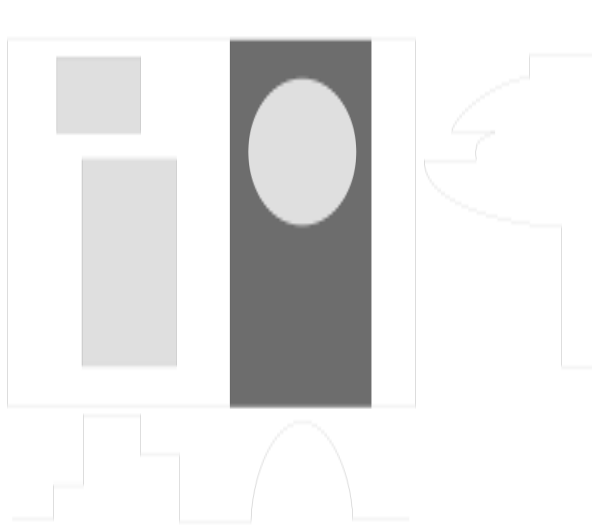


Figura 7.19. Histogramas laterales.

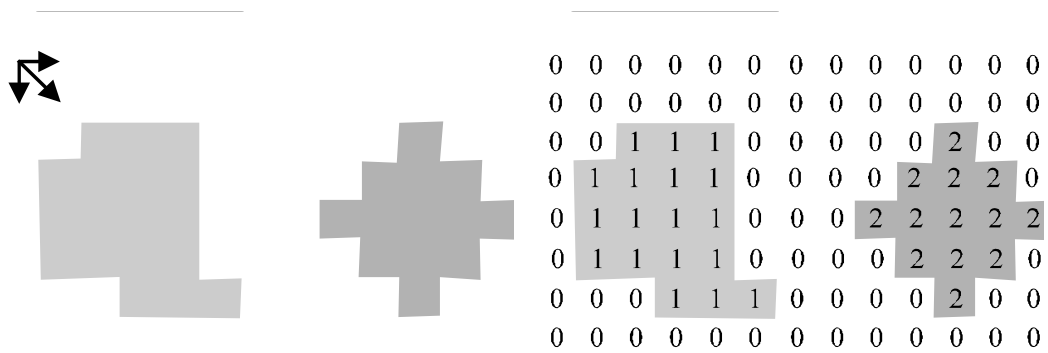


Figura 7.20. Algoritmo de etiquetado

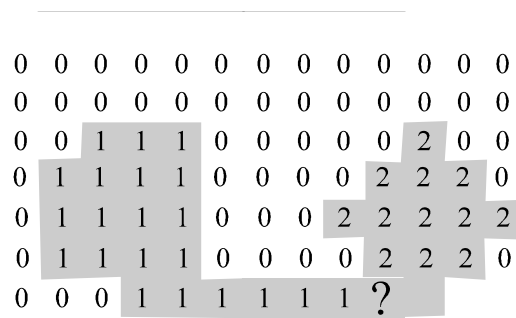


Figura 7.21. Resolución de ambigüedades

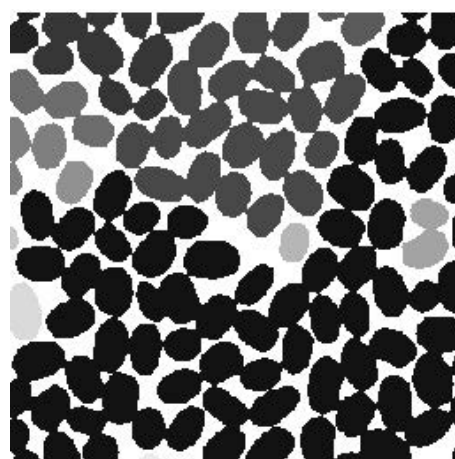
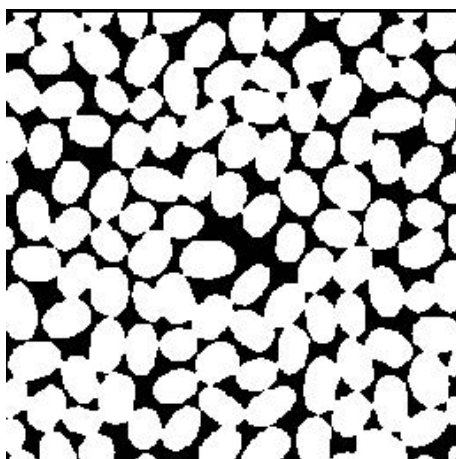


Figura 7.22. Etiquetado.



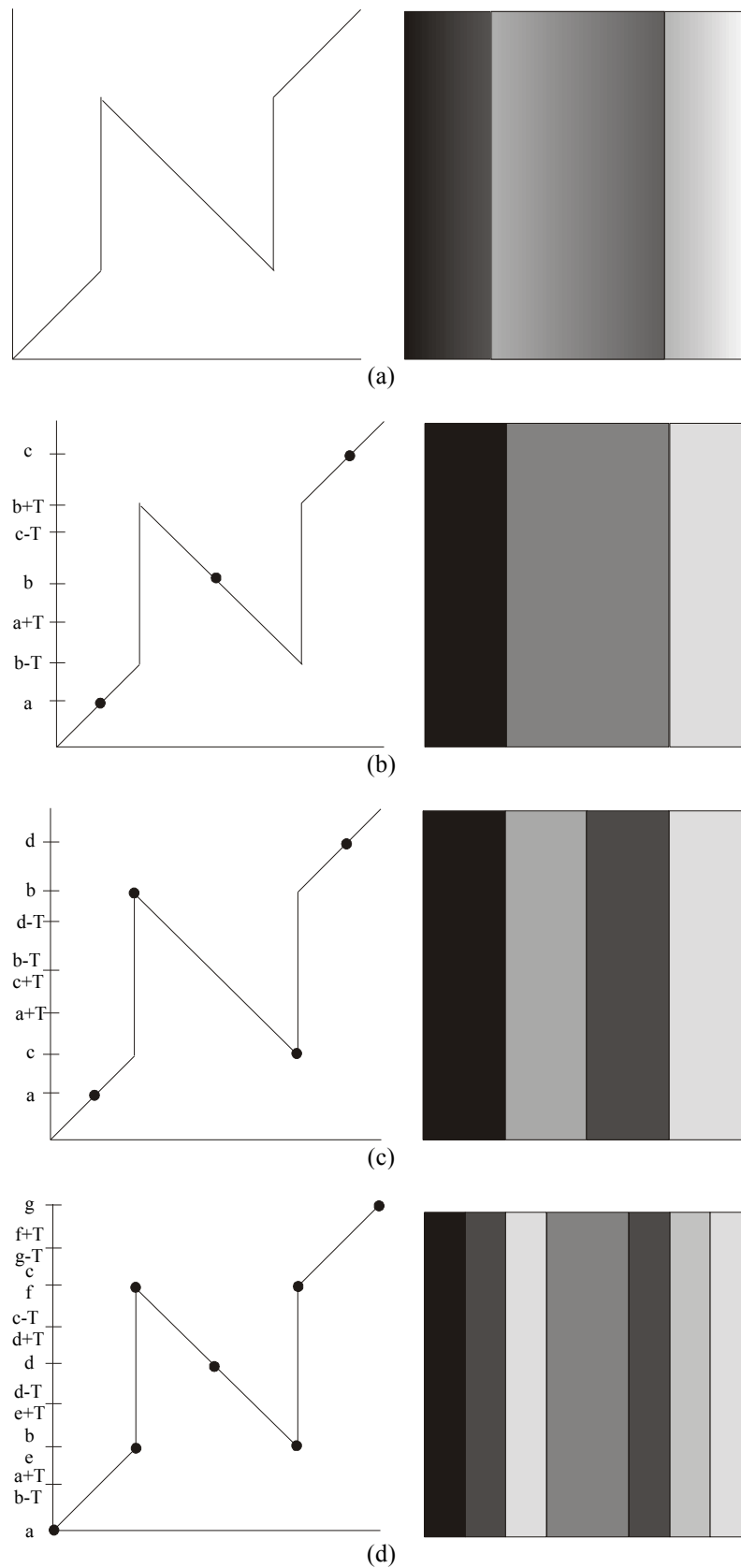


Figura 7.23. Elección de puntos semilla y umbrales en el crecimiento de regiones.

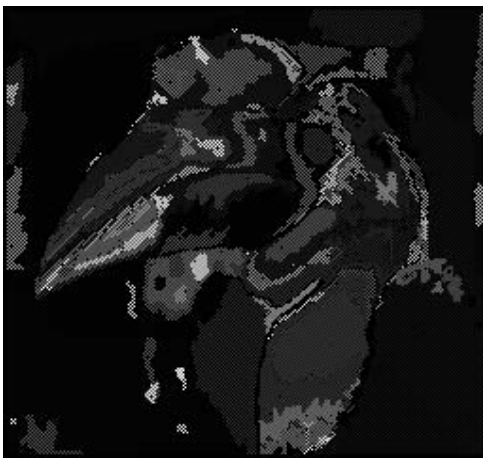


Figura 7.24. Resultado de aplicar el crecimiento de regiones

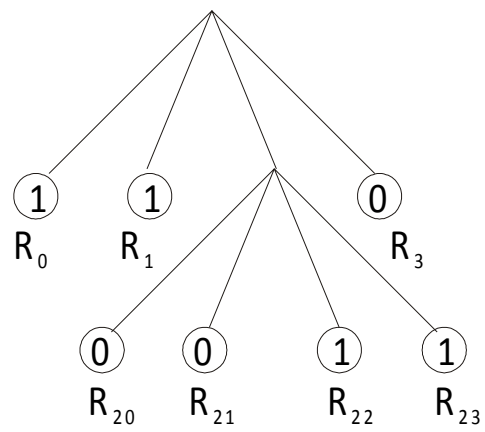
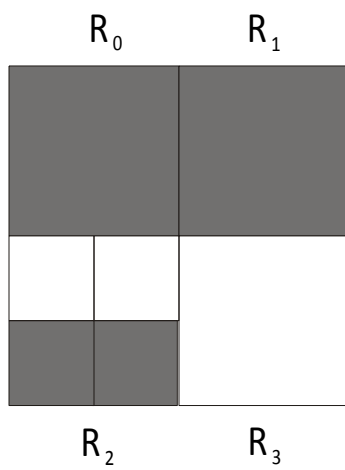


Figura 7.25 Estructura Quadtree.

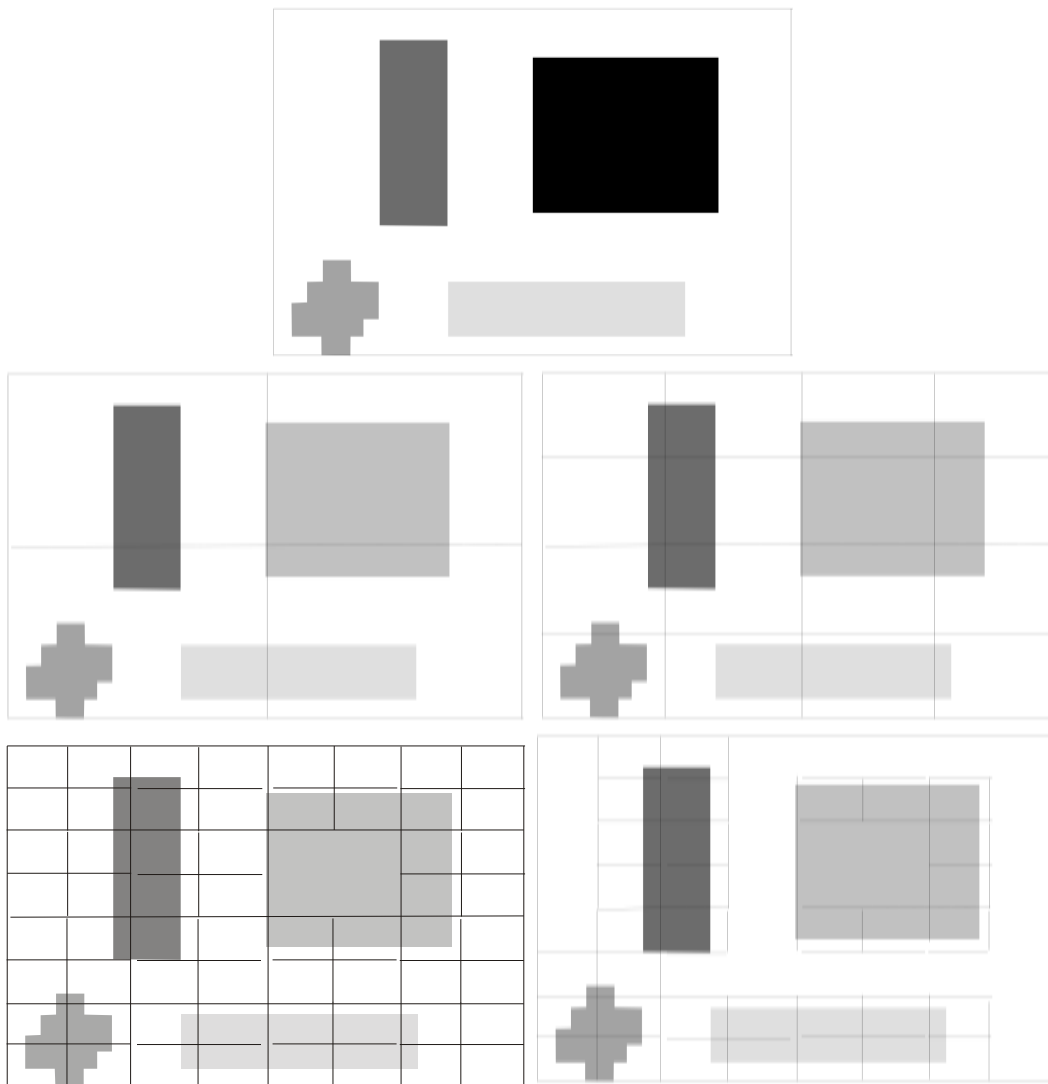


Figura 7.26. Algoritmo Split & Merge

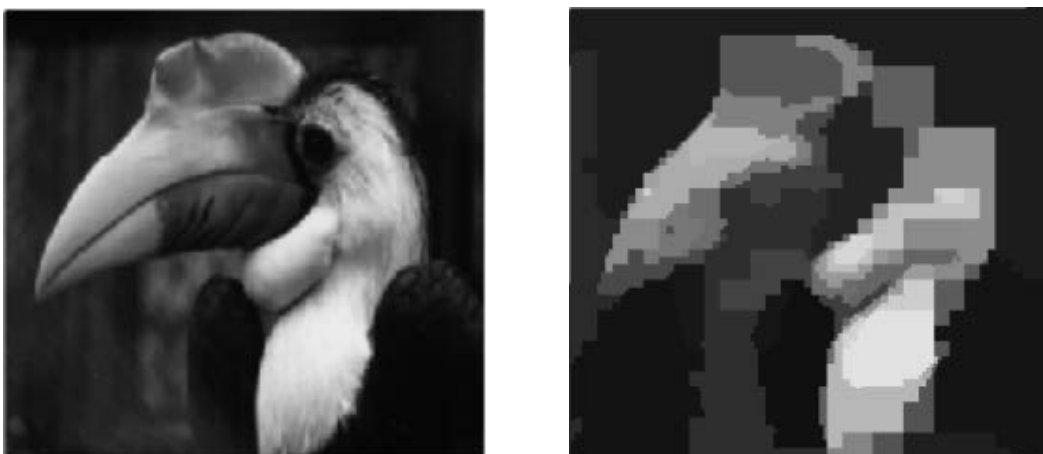


Figura 7.27. Resultado del Split & Merge

