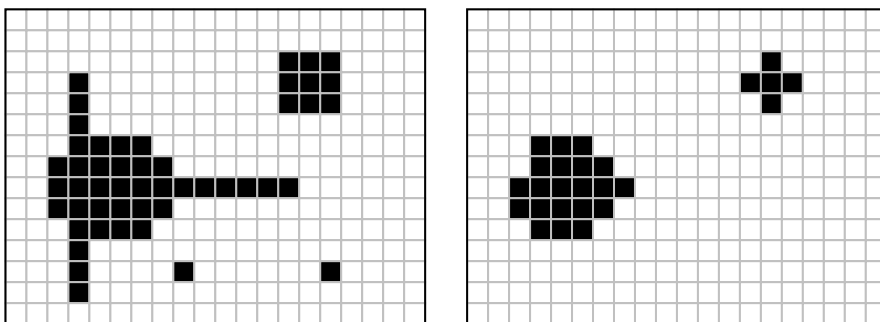


## OPERACIONES MORFOLOGICAS

En el capítulo 5 cuando fue tratado el filtro de la mediana pudo observarse como que este tipo de filtro tiene la capacidad de modificar las estructuras bidimensionales presentes en la imagen. Dentro de estos cambios pueden citarse, la modificación de las esquinas de estructuras, redondeándolas o bien el eliminar puntos o estructuras delgadas como líneas o pequeños artefactos (véase figura 9.1). El filtro de la mediana por lo anterior puede decirse que reacciona selectivamente a la forma de las estructuras locales de la imagen. Sin embargo la operación de este filtro no puede ser utilizada controladamente, es decir, no puede ser usado para reaccionar con estructuras particularmente descritas de acuerdo a su forma o estructura. En este capítulo se analizan los llamados filtros morfológicos, los cuales son capaces de influir sobre estructuras de la imagen, designadas previamente mediante la definición de elemento estructural.

Los filtros morfológicos fueron originalmente concebidos para ser usados sobre imágenes binarias, es decir sobre imágenes cuyos píxeles solo tienen dos posibles valores 1 y 0, blanco y negro. Las imágenes binarias se encuentran en un gran número de aplicaciones especialmente en procesamiento de documentos. En este capítulo tratará de igual manera sobre la aplicación y manipulación de los filtros morfológicos aplicados a imágenes en escala de grises.

En lo sucesivo definiremos a los píxeles de la estructura como los correspondientes a 1 y a los píxeles del fondo a los correspondientes a cero.



(a)

(b)

Figura 9.1 Filtro de la mediana 3x3 utilizado sobre una imagen binaria. (a) Imagen original y (b) resultado después de la acción del filtro.

## 1.1 CONTRACCIÓN Y CRECIMIENTO DE ESTRUCTURAS.

Partiendo del hecho comentado anteriormente, el filtro de la mediana 3x3 puede aplicarse sobre imágenes binarias para el redondeo de grandes estructuras y la eliminación de las pequeñas. Con ello puede ser utilizado el filtro de la mediana para eliminar estructuras cuyo tamaño es más pequeño al definido por el filtro de la mediana (en el caso del ejemplo 3x3).

De todo lo tratado surge la pregunta: ¿Cómo puede mediante la definición del tamaño y forma de una estructura utilizarse alguna operación que controladamente influya sobre estructuras dentro de una imagen?

Los efectos del filtro de la mediana pueden ser considerados como interesantes, sin embargo, su efecto sobre la imagen si bien es capaz de eliminar estructuras pequeñas también lo es el de afectar a las estructuras grandes. Otro enfoque que puede ser considerado mejor y además con la capacidad de afectar controladamente a las estructuras de la imagen se basa en el siguiente principio (véase figura 9.2):

1. Primero todas las estructuras de la imagen son reducidas, de lo cual resulta que estructuras pequeñas son sencillamente eliminadas.
2. A través del encogimiento, estructuras pequeñas son eliminadas mientras que las grandes permanecen.
3. Después, las estructuras reducidas son aumentadas, hasta que las estructuras grandes recuperen su tamaño original.

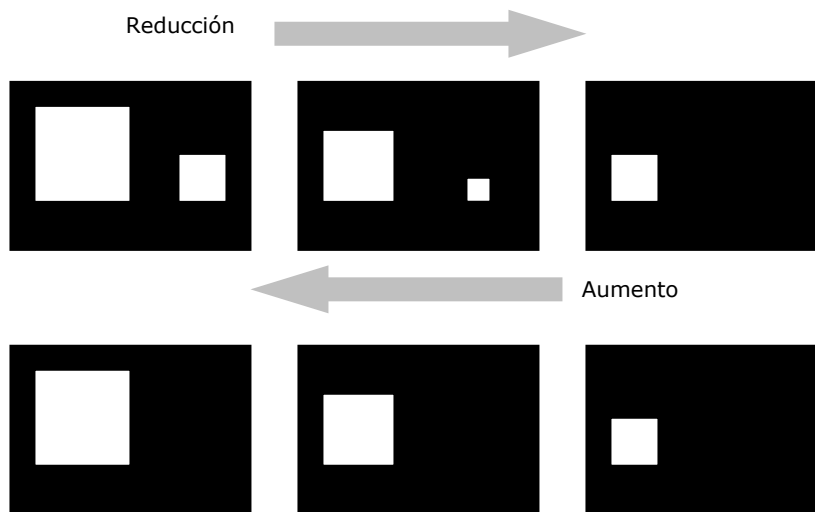


Figura 9.2 Por medio de la operación sucesiva de reducciones y posteriormente de aumentos de estructuras en una imagen, es posible eliminar elementos pequeños.

De la figura 9.2 es evidente que para la eliminación de estructuras pequeñas solo es necesario definir 2 diferentes operaciones: **Reducción** la cual mediante su aplicación permite que los píxeles de la última capa del objeto que se encuentra en contacto con los píxeles del fondo sean eliminados (figura 9.3). Lo contrario sucede con la operación **Aumento**, donde se añade una capa de píxeles (que pertenecen al fondo) a la estructura, de tal forma que crece en tamaño (Figura 9.4).

### 1.1.1 Tipos de vecindad entre píxeles.

En ambos tipos de operaciones definidas intuitivamente es necesario determinar, la forma en que dos píxeles establecen una relación de vecindad entre ellos. Comúnmente pueden diferenciarse dos tipos de vecindad.

- **Vecindad 4.** Bajo este enfoque se considera que existe vecindad si un píxel se encuentra en relación directa con algún píxel vecino que se encuentre arriba, abajo, del izquierdo o del lado derecho del píxel en cuestión (Figura 9.5(a)).
- **Vecindad 8.** Bajo este enfoque se considera que existe vecindad si un píxel además de las relaciones especificadas por la vecindad 4, presenta también relación directa con los píxeles diagonales a el (Figura 9.5(b)).

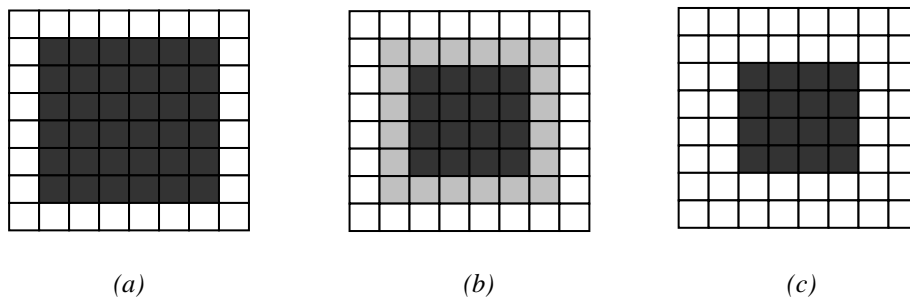


Figura 9.3 Reducción de una región de la imagen a través de la eliminación de la frontera de píxeles de la estructura. (a) Imagen original, (b) realce de los píxeles de la frontera de la estructura y (c) eliminación de los píxeles frontera de la estructura.

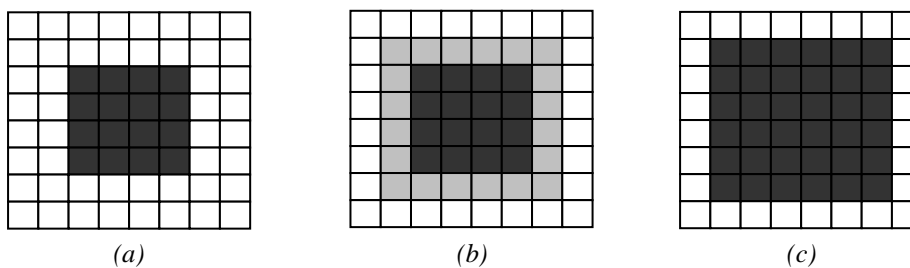


Figura 9.4 Aumento de una región de la imagen a través del crecimiento de la frontera de píxeles de la estructura que pertenecen al fondo. (a) Imagen original, (b) realce de los píxeles de la frontera de la estructura que pertenecen al fondo (c) aumento de los píxeles frontera de la estructura que pertenecen al fondo.



(a)

(b)

Figura 9.5 Definición de la vecindad de un píxel P. (a) Vecindad 4, definida sobre P y (b) Vecindad 8, definida sobre P.

## 1.2 OPERACIONES MORFOLÓGICAS FUNDAMENTALES

Las operaciones de reducción y crecimiento son dos de las operaciones morfológicas fundamentales desempeñadas por filtros, las cuales guardan una relación cercana con los procesos físicos de **erosión** y **dilatación**. Entre ambos procesos y los descritos cualitativamente en la sección pasada existe una relación interesante, ya que si se concibe a la dilatación, como un crecimiento y a la erosión como un desgaste caracterizado por la pérdida de área o volumen, serian perfectas analogías de las ideas propuestas. De lo anterior, podría decirse que el añadir una capa de píxeles a una estructura en una imagen, se dilataría la estructura, mientras que si se elimina una capa de píxeles de una estructura, sería erosionada.

### 1.2.1 La estructura de referencia

De igual manera que para la operación de un filtro es necesario la definición de una matriz de coeficientes, para la operación de un filtro morfológico se necesita la caracterización de una matriz, llamada estructura de referencia. Al igual que una imagen binaria, la estructura de referencia contiene también sólo elementos 0 y 1. Por lo que

$$H(i, j) \in \{0, 1\} \quad (9.1)$$

y posee, al igual que la matriz de coeficientes de los filtros, su propio sistema de coordenadas, teniendo al punto referencial (véase figura 9.6) como origen<sup>1</sup>.

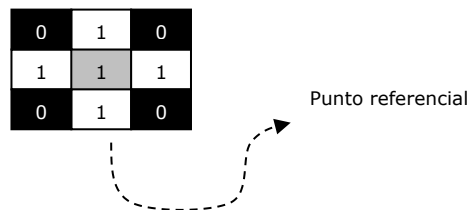


Figura 9.6 Ejemplo de una estructura de referencia para operaciones morfológicas binarias.

### 1.2.2 Conjunto de puntos

Para la descripción formal de las operaciones morfológicas resulta más práctico describir a las imágenes como conjuntos, cuyos elementos pueden ser referenciados mediante coordenadas

<sup>1</sup> Llamado referencial por que no necesariamente es el centro, a diferencia de la matriz de coeficientes de un filtro.

bidimensionales. Por ejemplo, una imagen binaria  $I(x, y)$  consiste de un conjunto de puntos  $P_I$  cuyos pares de coordenadas corresponden a los puntos cuyo valor de píxel es 1, lo cual puede ser definido como:

$$P_I = \{(x, y) | I(x, y) = 1\} \quad (9.2)$$

Como se muestra en la figura 9.7, puede ser descrita mediante la notación de conjuntos definida anteriormente no solamente una imagen sino también una estructura de referencia.

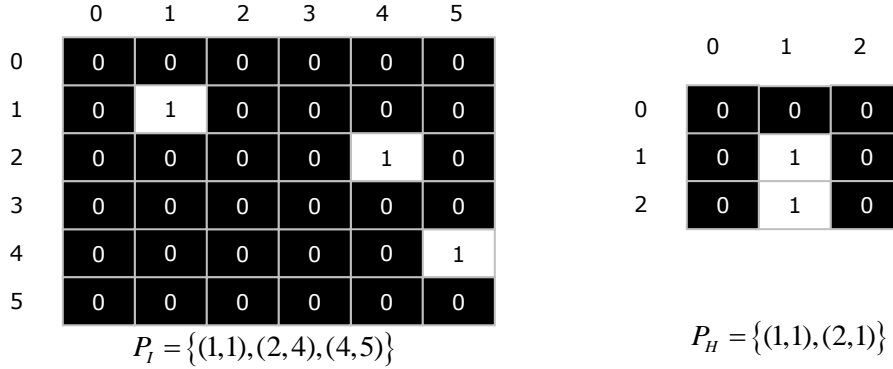


Figura 9.7 Descripción de una imagen binaria  $I$  y una estructura de referencia  $H$  como conjuntos de pares de coordenadas  $P_I$  y  $P_H$ .

Mediante esta descripción pueden ser definidas de forma sencilla las operaciones que se realizan sobre imágenes binarias. Por ejemplo, la inversión de una imagen binaria que intercambia el valor de los píxeles de 1 a 0 y viceversa puede definirse como el conjunto complemento, tal que

$$P_{\neg I} = \bar{P}_I \quad (9.3)$$

Donde  $\neg I$  define la inversión de  $I$ . De igual manera si dos imágenes binarias  $I_1$  e  $I_2$  son elemento a elemento, unidos a través de la función lógica OR, entonces la definición de esta operación en formato de conjuntos puede expresarse como:

$$P_{I_1 \vee I_2} = P_{I_1} \cup P_{I_2} \quad (9.4)$$

Debido a que la notación de conjunto de puntos es una representación alternativa de operaciones binarias, se utilizara de forma alternada la notación, atendiendo a la economía de la descripción y entendimiento de la operación, dependiendo del caso. Por ejemplo, significará lo mismo la operación  $I_1 \cup I_2$  que  $P_{I_1} \cup P_{I_2}$  o bien  $\bar{I}$  que  $\bar{P}_I$ .

### 1.2.3 Dilatación.

La dilatación es una operación morfológica correspondiente a la idea intuitiva del crecimiento, o de añadir una capa de píxeles a la estructura de la imagen. La manera específica en la que es controlado el crecimiento, es por medio de una figura referida como estructura de referencia. Esta operación es definida en la notación de conjuntos de puntos como:

$$I \oplus H = \{(x', y') = (x + i, y + j) | (x', y') \in P_I, (i, j) \in P_H\} \quad (9.5)$$

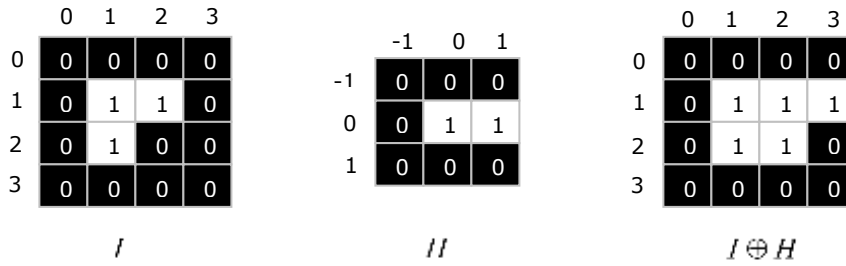
Como indica la expresión 9.5, el conjunto de puntos que constituyen a la dilatación de una imagen  $I$  y la estructura de referencia  $H$ , queda definido por todas las posibles combinaciones de los pares de coordenadas de los conjuntos de puntos  $P_I$  y  $P_H$ . Se podría también interpretar a la operación de dilatación, como el resultado de añadir a los píxeles 1 de la imagen ( $P_I$ ) la forma correspondiente a la estructura de referencia. Lo anterior puede ser ilustrado en la figura 9.8.

### 1.2.4 Erosión.

La operación quasi-inversa de la operación dilatación es la erosión, la cual expresada en la notación de conjunto de puntos es definida como:

$$I \ominus H = \{(x', y') | (x' + i, y' + j) \in P_I, \forall (i, j) \in P_H\} \quad (9.6)$$

Esta expresión dice que para cada punto  $(x', y')$  de la imagen, el resultado se compone por los puntos  $(x', y')$  para los cuales todos los posibles valores  $(x' + i, y' + j)$  se encuentran en  $I$ . La figura 9.9 muestra un ejemplo de la erosión desempeñada entre una imagen  $I$  y la estructura de referencia  $H$ .



$$I \oplus H = \{(1,1) + (0,0), (1,1) + (0,1), (1,2), (0,0) + (1,2) + (0,1), (2,1) + (0,0), (2,1) + (0,1)\}$$

Figura 9.8 Ejemplo de dilatación. A la imagen binaria  $I$  se le aplica la operación de dilatación, utilizando como estructura de referencia  $H$ . El elemento estructural  $H$  es añadido en cada píxel 1 de la imagen  $I$ .

Este proceso puede ser interpretado de la siguiente manera: Un píxel  $(x', y')$  resultado de la erosión es uno si la estructura de referencia centrada en este píxel coincide en forma con el contenido de la imagen. La figura 9.9 muestra como la estructura de referencia centrada en el píxel (1,1) de la imagen coincide en forma con el contenido de la imagen, por lo que el resultado de la erosión para este píxel es 1.

### 1.2.5 Propiedades de la dilatación y erosión

Las operaciones de dilatación y erosión no pueden ser consideradas en el sentido estricto inversas, debido a que no es posible reconstruir completamente una imagen erosionada, mediante la aplicación sucesiva de dilataciones, lo mismo aplica en sentido inverso. Sin embargo presentan una estrecha relación dual, ya que una dilatación de los píxeles con valor de 1 de una imagen puede llevarse a cabo realizando una erosión del fondo o de los píxeles en cero. Esto puede ser expresado como:

$$\overline{I \ominus H} = \overline{I} \oplus \overline{H} \quad (9.7)$$

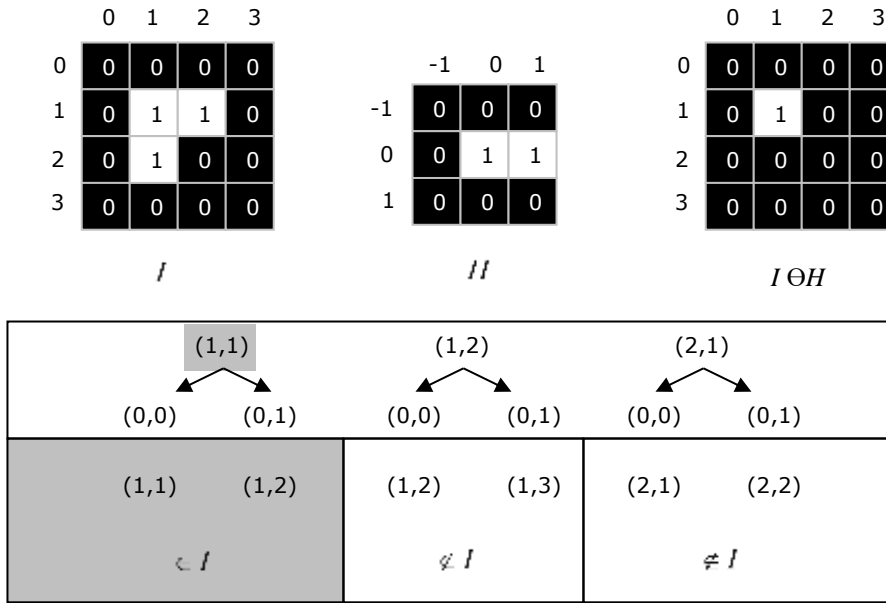


Figura 9.9 Ejemplo de erosión. La imagen  $I$  es erosionada a través de la estructura de referencia  $H$ . Del proceso ilustrado puede verse como el píxel de  $I$  con coordenadas  $(1,1)$  al sumarse con los elementos  $(0,0)$  y  $(0,1)$  de la estructura de referencia se obtienen los puntos  $(1,1)$  y  $(1,2)$  los cuales forman parte de la imagen  $I$ , por lo que el resultado de la erosión será uno en la posición  $(1,1)$ . Para los demás píxeles de la imagen no se cumple lo anterior.

La dilatación es también conmutativa, por lo que es válido decir:

$$I \oplus H = H \oplus I \quad (9.8)$$

Por lo tanto al igual que en la convolución, es posible intercambiar la imagen y la estructura de referencia. De la propiedad anterior se desprende que la dilatación es de igual forma una operación asociativa, debido a eso es válido decir

$$(I_1 \oplus I_2) \oplus I_3 = I_1 \oplus (I_2 \oplus I_3) \quad (9.9)$$

De lo anterior puede decirse que la secuencia en que se realizan las operaciones entre los diferentes argumentos no es relevante. Esta propiedad, es de utilidad, ya que de esta manera la estructura de referencia puede ser dividida en estructuras más pequeñas, de tal manera que la

velocidad de la operación completa es mayor, al haber un número más pequeño de operaciones. Considerando lo anterior podría ser expresada la operación de dilatación como:

$$I \oplus H = ((I \oplus H_1) \oplus H_2) \quad (9.10)$$

Donde  $H_1$  y  $H_2$  corresponden a las estructuras de referencia en las que fue dividida  $H$ .

La erosión, sin embargo, no es conmutativa, por lo que:

$$I \ominus H \neq H \ominus I \quad (9.11)$$

### 1.2.6 Diseño de filtros morfológicos

Los filtros morfológicos se especifican mediante la definición de dos cosas, la operación que desempeñan (erosión o dilatación) y su correspondiente estructura de referencia. El tamaño y la forma de la estructura de referencia son dependientes de la aplicación. En la práctica son frecuentemente utilizadas las estructuras de referencia mostradas en la figura 9.10. Una estructura de referencia en forma de disco con un radio  $r$  añade una capa de  $r$  píxeles a los objetos de la imagen cuando es utilizada a través de la operación de dilatación. El efecto contrario sucede cuando la misma estructura es utilizada con la operación de erosión, por lo que en este caso una capa de  $r$  píxeles es extraída de los objetos de la imagen. Teniendo como imagen original la mostrada en la figura 9.11(a) y como estructura de referencia la mostrada en 9.11(b), se obtiene como resultados de la dilatación y erosión de la imagen, los mostrados en la figura 9.12, considerando diferentes valores de  $r$ .

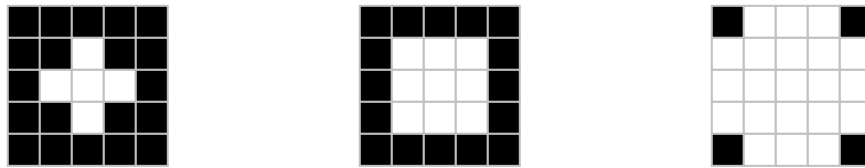


Figura 9.10 Estructuras de referencia típicas de diferentes tamaños. (a) estructura de 4 vecinos, (b) estructura de 8 vecinos y (c) estructura de disco pequeño.

Otros resultados de las operaciones morfológicas de erosión y dilatación para estructuras de referencia de diferentes formas son mostrados en la figura 9.13.

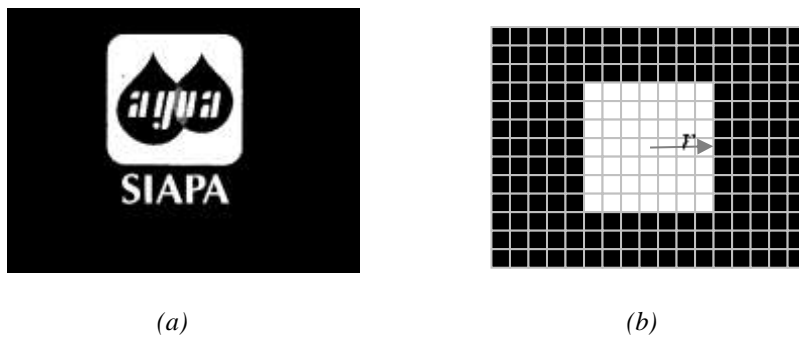


Figura 9.11 (a) Imagen original y (b) estructura de referencia utilizados en los ejemplos que ilustran el efecto en las operaciones de dilatación y erosión ante un valor diferente de  $r$ , donde  $r$  define el tamaño del disco.



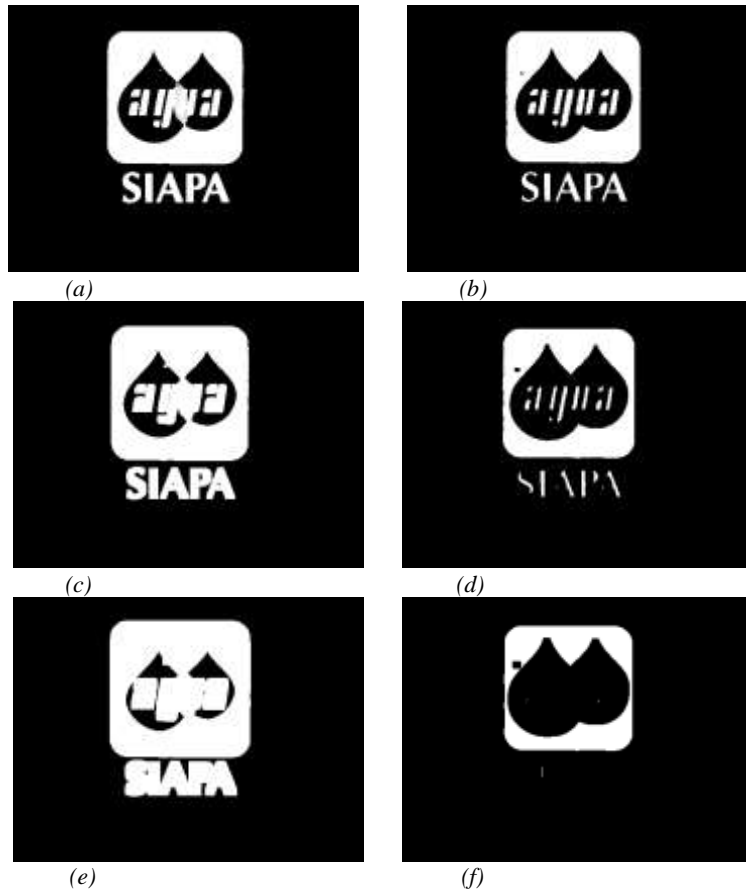


Figura 9.12 (a) Dilatación y (b) erosión considerando  $r=2$ , (c) dilatación y (d) erosión considerando  $r=5$  y (e) dilatación y (f) erosión considerando  $r=10$ .

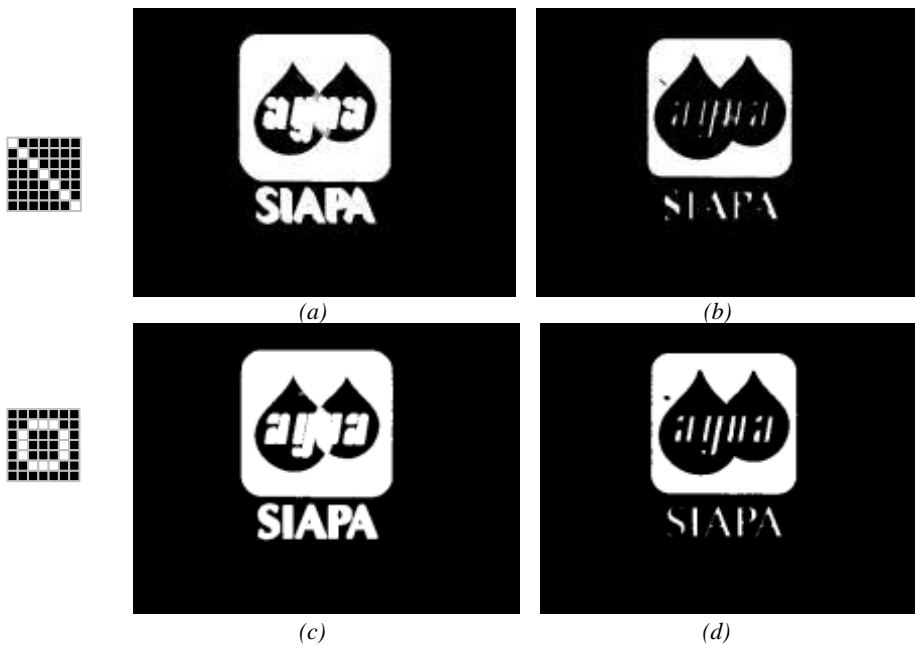


Figura 9.13 Dilatación y erosión binarias realizadas sobre la imagen ejemplo y utilizando como estructura de referencia la mostrada al lado izquierdo. (a) Dilatación y (b) erosión usando como estructura de

referencia valores de 1 en la diagonal principal y (c) dilatación y (d) erosión usando como estructura de referencia un patrón circular.

A diferencia de los filtros espaciales (Capítulo 5), es en general imposible crear filtros isotrópicos<sup>2</sup> en dos dimensiones  $H$  a partir de estructuras de referencia unidimensionales  $H_x$  y  $H_y$ . Lo anterior se debe a que la operación morfológica que se establezca entre ambas estructuras de referencia da como origen siempre una nueva estructura de referencia cuadrada y por lo tanto no isotrópica.

El método más usado para la implementación de filtros morfológicos de grandes dimensiones es la aplicación repetitiva de una misma estructura de referencia de dimensiones pequeñas, de tal forma que el resultado es de nueva cuenta usado para aplicarle la operación con la misma estructura de referencia. Lo anterior arrojaría aproximadamente el mismo resultado que si se utilizara una estructura de referencia de grandes dimensiones (véase figura 9.14).

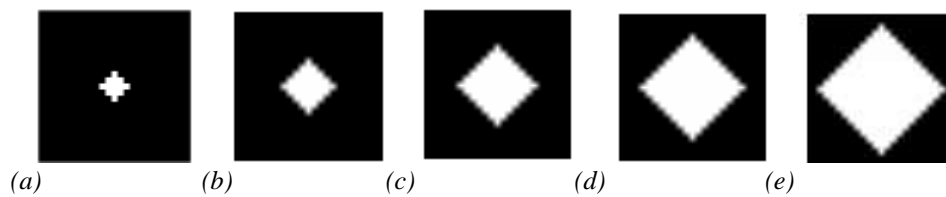


Figura 9.14 Formación de filtros de grandes dimensiones mediante la aplicación repetida de los operadores sobre la imagen. (a)  $H$ , (b)  $H \oplus H$ , (c)  $H \oplus H \oplus H$ , (d)  $H \oplus H \oplus H \oplus H$  y (e)  $H \oplus H \oplus H \oplus H \oplus H$ .

### 1.3 DETECCIÓN DE BORDES EN IMÁGENES BINARIAS

Una aplicación típica de las operaciones morfológicas es la extracción de los bordes de los objetos contenidos en una imagen binaria. El proceso para la detección, el cual es descrito en el algoritmo 9.1, comienza con aplicar la operación morfológica de erosión (1) sobre la imagen, utilizando como estructura de referencia alguna de las definidas en la figura 9.10, las cuales corresponden a relaciones de vecindad. El objetivo de aplicar la erosión es el de eliminar de la imagen original la capa exterior (borde) de los objetos dentro de la imagen.

Al considerar lo anterior, el borde de los objetos de la imagen puede encontrarse si, a la imagen que se erosionó se obtiene su inversa (2), con ello se tendrá una imagen donde los objetos tendrán un valor de cero, mientras que el fondo de la imagen más los bordes de los objetos están en uno. Por lo que si se hace una operación AND entre la imagen original y la inversa (3) de la versión erosionada de la imagen, se obtendrán los bordes de los objetos, ya que los únicos píxeles comunes entre ambas imágenes son los bordes. La figura 9.15 muestra la detección de bordes usando el algoritmo 9.1.

### 1.4 COMBINACIÓN DE OPERACIONES MORFOLÓGICAS

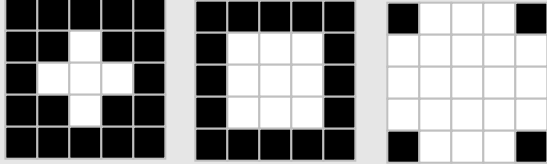
En aplicaciones prácticas de procesamiento de imagen la dilatación y erosión son usadas en la mayoría de los casos en varias combinaciones. En esta sección se considera tres de las combinaciones más comunes de la erosión y dilatación tales como apertura, cerradura y la transformación éxito o fracaso ('hit or miss'). Debido a la quasi-dualidad existente entre las

<sup>2</sup> Invariantes a la rotación

operaciones morfológicas de dilatación y erosión, pueden ser utilizadas en combinación. De las combinaciones existentes dos son de especial importancia, las cuales tienen nombres y símbolos especiales. Dichas operaciones son llamadas apertura (*'Opening'* en inglés) y cierre (*'closing'*).

**Detección de bordes por erosión ( $I_b(x, y)$ )**  
 Donde  $I_b(x, y)$  es una imagen binaria

**1:** Se hace erosión de  $I_b(x, y)$  mediante la utilización de una de las siguientes estructuras de referencia obteniéndose  $I'$ .

$$I'(x, y) = I \ominus H$$


**2:** Se invierte la imagen que se erosiona.

$$I_{inv}(x, y) = \overline{I'(x, y)}$$

**3:** Se aplica la operación AND entre  $I_{inv}(x, y)$  e  $I_b(x, y)$

$$I_{borde}(x, y) = I_{inv}(x, y) \vee I_b(x, y)$$

Algoritmo 9.1 Algoritmo de detección de bordes usando operaciones morfológicas.

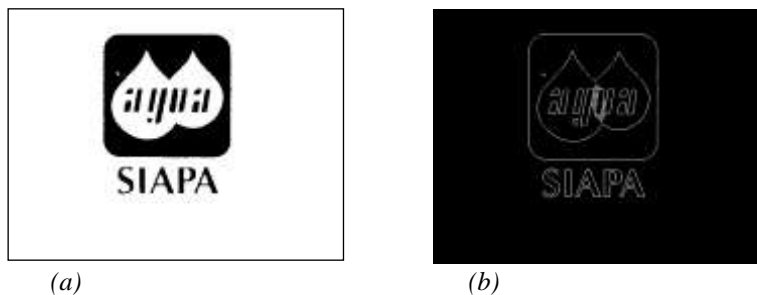


Figura 9.15 Detección de bordes usando la operación morfológica de erosión. (a) Imagen  $I_{inv}(x, y)$  obtenida del algoritmo 9.1 considerando como imagen original la definida en la figura 9.11(a) y (b) imagen de los bordes  $I_{borde}(x, y)$  obtenida después de ejecutar el algoritmo 9.1.

### 1.4.1 Apertura

La apertura es definida como una erosión seguida por una dilatación, utilizando para ello la misma estructura de referencia para ambas operaciones. Lo anterior es definido como:

$$I \circ H = (I \ominus H) \oplus H \quad (9.12)$$

Una apertura produce, en su primera parte (erosión), que todos los píxeles de la imagen en uno, que sean más pequeños que la estructura de referencia, sean eliminados. Las estructuras que permanezcan en la imagen serán, a través de la dilatación, suavizadas y aumentadas aproximadamente igual a su tamaño original (antes de aplicar la erosión). La figura 9.16 muestra el efecto de la

operación de apertura utilizando una estructura de referencia en forma de disco al considerar diferentes tamaños del disco ( $r$ , véase figura 9.16(b)).

### 1.4.2 Cierre

A la secuencia de operaciones de dilatación seguida de la erosión, usando para ello la misma estructura de referencia, se denomina cierre. Dicha operación es definida como:

$$I \bullet H = (I \oplus H) \ominus H \quad (9.13)$$

A través de la operación de cierre, se rellenan los hoyos encontrados dentro de los objetos de la imagen y que además son más pequeños que la estructura de referencia. La figura 9.16 muestra el efecto de la operación de cierre utilizando una estructura de referencia en forma de disco al considerar diferentes tamaños del disco ( $r$ , véase figura 9.16(b)).

### 1.4.3 Propiedades de las operaciones Apertura y Cierre.

Ambas operaciones de apertura y cierre son *idempotentes*, esto significa que se podría realizar la operación varias veces y aún así conseguir el mismo resultado que se obtendría si se realizase una sola vez, esto es:

$$\begin{aligned} (I \circ H) \circ H &= I \circ H \\ (I \bullet H) \bullet H &= I \bullet H \end{aligned} \quad (9.14)$$

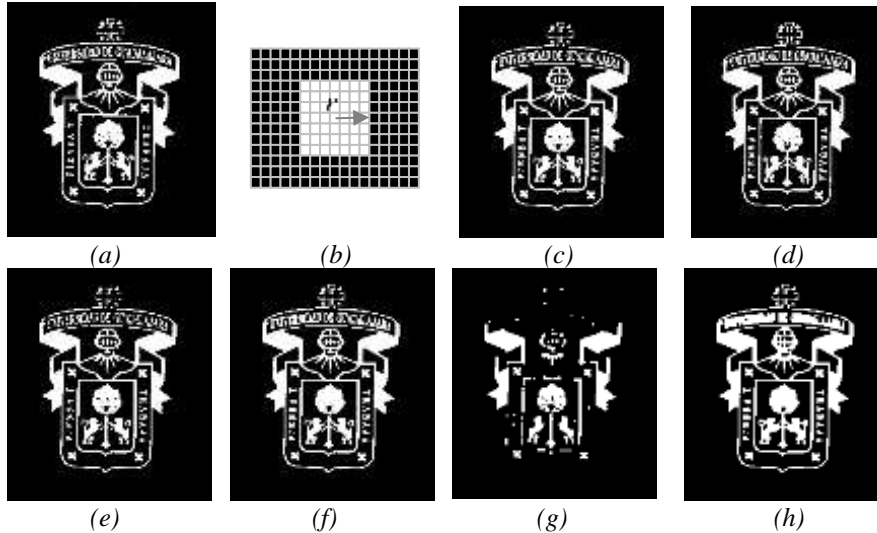


Figura 9.16 Efectos de las operaciones de apertura y cierre ante diferentes tamaños de la estructura de referencia. (a) Imagen original, (b) estructura de referencia y el parámetro  $r$  que modifica su tamaño, (c) operación de apertura considerando  $r=1$ , (d) operación de cierre con  $r=1$ , (e) apertura para  $r=3$ , (f) cierre para  $r=3$ , (g) apertura para  $r=5$  y (h) cierre para  $r=5$ .

Otra propiedad importante que poseen ambas operaciones es la *dualidad*, en el sentido de que una apertura de los valores de la imagen que están en uno (1) es equivalente al cierre de los valores de la imagen en cero (0). Esto definido sería:

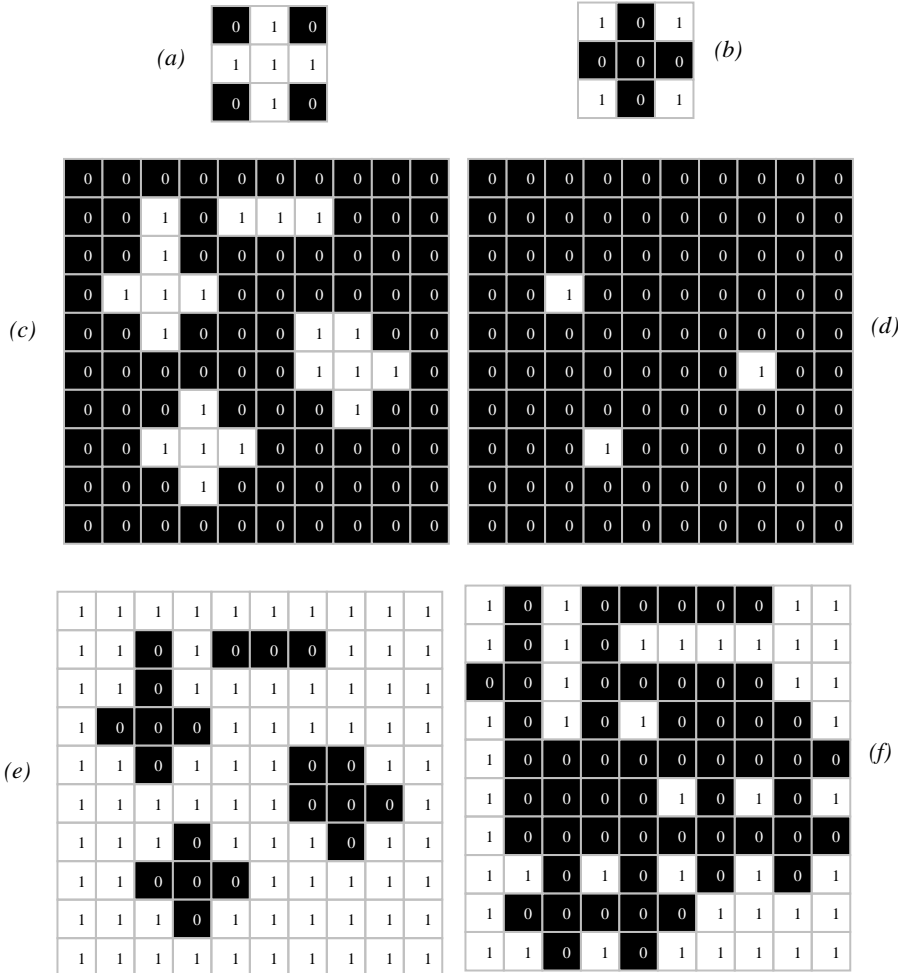
$$\begin{aligned} I \circ H &= \overline{(\bar{I} \bullet H)} \\ I \bullet H &= \overline{(\bar{I} \circ H)} \end{aligned} \quad (9.15)$$

#### 1.4.4 La transformación de éxito o fracaso ('Hit-or-miss')

La transformación de éxito o fracaso es útil para la identificación de determinadas configuraciones de píxeles. La transformada de éxito o fracaso de  $I$  y  $H$  es denotada como  $I \otimes H$ . En la expresión,  $H$  es una estructura de referencia constituida por un par  $H = (H_1, H_2)$  de estructuras de referencia, en lugar de una sola, tal y como fue utilizado para las operaciones de apertura y cierre. La transformada de éxito o fracaso es definido en términos de dos estructuras de referencia como:

$$I \otimes H = (I \ominus H_1) \cap (I^c \ominus H_2) \quad (9.16)$$

La figura 9.17 y 9.18 muestra la forma en la que puede ser usada la transformación de éxito o fracaso para la localización de configuraciones de píxeles en una imagen. En este caso la configuración a detectar es una cruz de píxeles.



(e)

(f)

Figura 9.17 Ilustración del proceso de la utilización de la transformada de éxito o fracaso. (a) estructura de referencia  $H_1$ , (b) estructura de referencia  $H_2$ , (c) imagen original, (d) resultado de  $I \ominus H_1$ , (e) imagen original invertida  $I^c$  y (f) resultado de  $I^c \ominus H_2$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Figura 9.18 Resultado de la transformada de éxito o fracaso  $I \otimes H = (I \ominus H_1) \cap (I^c \ominus H_2)$  de la Fig. 9.17.

De la operación final, la intersección ( $\cap$ ) representa la operación lógica AND. El nombre de “éxito o fracaso” que tiene la transformada, proviene del hecho de las operaciones que se desarrollan en la operación. Es decir, en la primera erosión ( $H_1$ ), se detectan todas las posiciones que coinciden con la estructura de referencia (éxito), mientras que en la segunda erosión ( $H_2$ ) se detectan las posiciones donde no hay coincidencia (fracaso).

## 1.5 FILTROS MORFOLÓGICOS PARA IMÁGENES A ESCALA DE GRISES

Las operaciones morfológicas no se restringen a su uso con imágenes binarias, sino que de igual manera pueden ser definidas para imágenes a escala de grises. Todas las operaciones morfológicas anteriormente vistas con excepción de la transformación éxito o fracaso tienen extensiones naturales a imágenes en escala de grises. De igual manera, las operaciones morfológicas pueden operar con imágenes a color procesando cada plano como si se tratase de una imagen de intensidad independiente. A pesar de que las operaciones morfológicas se llaman igual y utilizan la misma simbología, su definición para las imágenes de intensidad varía fuertemente en comparación a las imágenes binarias.

### 1.5.1 Estructura de referencia

La primera diferencia existente entre las operaciones morfológicas para los diferentes tipos de imagen es la estructura de referencia. La estructura de referencia no representa más una matriz de unos y ceros que describen la forma y dimensión de la estructura que operara sobre las imágenes binarias, sino que ahora es una matriz de coeficientes “similar” a la definida para filtros espaciales. Considerando lo anterior la estructura de referencia se describe como una función bidimensional de valores reales tal que:

$$H(i, j) \in \mathbb{R} \quad (9.17)$$

Los valores de la estructura de referencia  $H(i, j)$  pueden ser positivos, negativos o bien cero. Sin embargo, a diferencia de la matriz de coeficientes de un filtro espacial, los ceros influyen en el resultado final del cálculo.

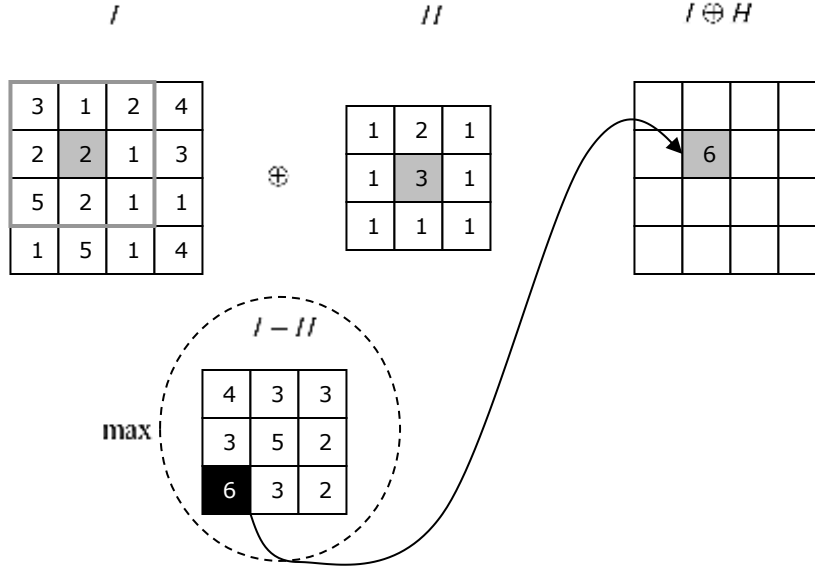


Figura 9.19 Dilatación sobre imágenes a escala de grises  $I \oplus H$ . La estructura de referencia 3x3 es centrada sobre la imagen  $I$ . Los valores de la imagen  $I$  son sumados elemento a elemento con los correspondientes valores de  $H$ , de la suma  $I + H$ , se elige el máximo, el cual es considerado como resultado de este proceso.

### 1.5.2 Dilatación y erosión para imágenes de intensidad

La dilatación para imágenes a escala de grises ( $\oplus$ ), es definida como el máximo de la suma realizada entre los valores de la estructura de referencia y la región de la imagen con la cual se corresponde. Esto es:

$$(I \oplus H)(x, y) = \max_{(i, j) \in H} \{I(x + i, y + j) + H(i, j)\} \quad (9.18)$$

Mientras que la erosión es definida como el mínimo de la resta realizada entre los valores de la estructura de referencia y la región de la imagen con la cual se corresponde. Lo anterior significa que:

$$(I \ominus H)(x, y) = \min_{(i, j) \in H} \{I(x + i, y + j) - H(i, j)\} \quad (9.19)$$

La figura 9.19 muestra mediante un ejemplo el efecto de la dilatación sobre imágenes a escala de grises. La figura 9.20 ilustra el resultado de haber aplicado la operación de erosión. En ambas operaciones puede ocurrir que existan valores fuera del rango normal usado para representar los datos de las imágenes (0-255); si eso sucede, simplemente se toma su valor límite, por lo que si el numero es negativo, se considera como resultado el valor de cero y si el valor de la operación es mayor a 255,

se considera como resultado 255 (esta operación de considerar los límites es más conocido como ‘Clamping’).

*Figura 9.20 Dilatación sobre imágenes a escala de grises  $I \ominus H$  La estructura de referencia 3x3 es centrada sobre la imagen  $I$ . Los valores de la imagen  $I$  son restados elemento a elemento a los correspondientes valores de  $H$ , de la resta  $I - H$ , se elige el mínimo, el cual es considerado como resultado de este proceso.*

La dilatación y erosión pueden ser combinadas para realizar una variedad de efectos. Por ejemplo, substraer la erosión de una imagen a su versión dilatada, encontrando de esta manera un tipo de “gradiente morfológico”. Definido de otra manera se tiene:

$$\partial M = (I \oplus H) - (I \ominus H) \quad (9.20)$$

Resultados de las operaciones morfológicas de dilatación y erosión sobre imágenes a escala de grises son mostrados en la Figura 9.21, considerando además el gradiente morfológico anteriormente propuesto.



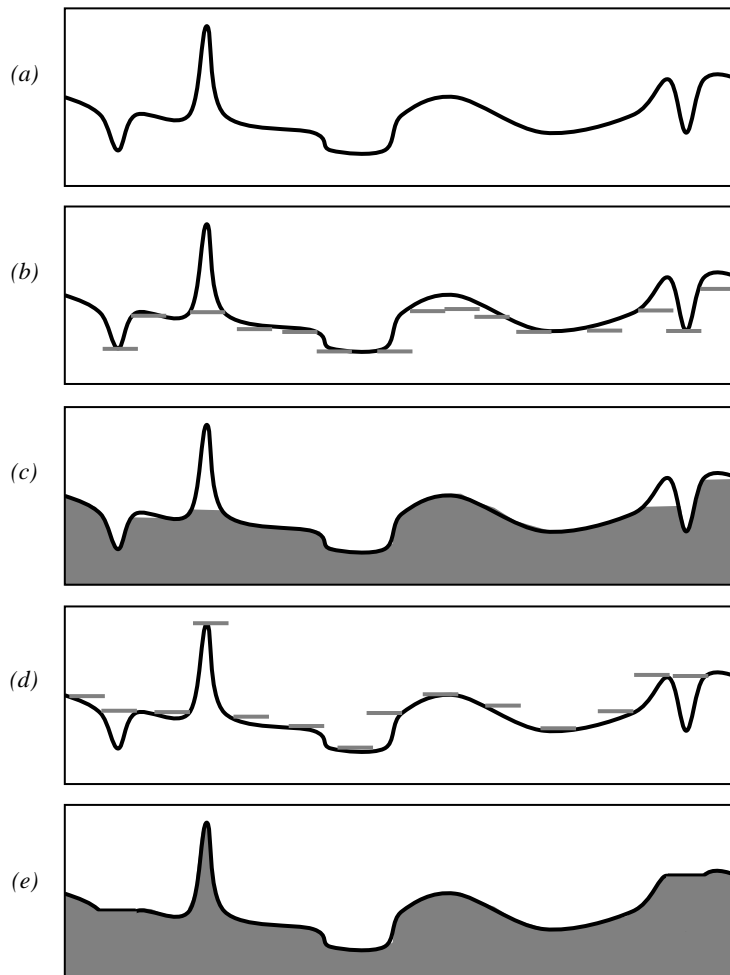




Figura 9.21 Operaciones morfológicas realizadas sobre imágenes a escala de grises. (a) Imagen original, (b) imagen erosionada, (c) imagen dilatada y (d) gradiente morfológico.

### 1.5.3 Operaciones de Apertura y cierre con imágenes de escala de grises

Las expresiones para la apertura y cierre de imágenes a escala de grises tienen la misma forma que sus contrapartes binarias. Ambas operaciones tienen una sencilla interpretación geométrica. La figura 9.22 muestra esta interpretación considerando un perfil de la imagen (una línea de la imagen). Para el caso de la operación de apertura, es como si a la curva de la figura 9.22(a) se le añadiera una estructura horizontal (fig. 9.22(b)) que empalme en la parte interior del trazo. Como puede ser visto el tamaño de la estructura no permite encajar dentro del pico presentado, de tal forma que en el resultado de la operación este máximo es eliminado (fig. 9.22(c)).





*Figura 9.22 Interpretación de las operaciones de apertura y cierre en imágenes a escala de grises. (a) perfil de una imagen a escala de grises, (b) aplicación de la estructura al perfil de acuerdo a la apertura, (c) resultado de la operación apertura, (d) aplicación de la estructura al perfil de acuerdo al cierre y (e) resultado de la operación cierre.*

En general, la operación de apertura es usada para remover pequeños picos de elementos iluminados en la imagen, mientras que los demás niveles de grises permanecen prácticamente inalterados.

La figura 9.22(d) da una ilustración gráfica de la operación de cierre. Nótese que a diferencia de la operación de apertura la estructura se coloca en el perfil en la parte de arriba, de tal forma que los valles si son más pequeños que la estructura, serán rellenados. La figura 9.22(e) muestra el resultado de la operación del cierre.

Contrario a la operación de apertura, la operación de cierre permite cancelar pequeños valores oscurecidos presentes en la imagen, los cuales son más pequeños que el tamaño de la estructura de referencia usada, dejando los demás valores de intensidad sin cambio significativo.

Debido a que la operación de apertura suprime detalles luminosos más pequeños que la estructura de referencia y a que la operación de cierre elimina artefactos oscuros, ambas operaciones son utilizadas combinadamente para suavizar y eliminar ruido en imágenes.

Al filtro producto de la combinación de la operación apertura y cierre es llamado **filtro de apertura y cierre**; dicho filtro tiene la posibilidad de eliminar simultáneamente artefactos brillantes y oscuros de imágenes, similares al ruido de sal y pimienta. En la operación del filtro, a la imagen a escala de grises se le aplica la operación de apertura (eliminando los artefactos brillantes) y al resultado de este proceso se le aplica después la operación de cierre (eliminando los artefactos oscuros). La figura 9.23 muestra el efecto de las operaciones apertura y cierre sobre una imagen a escala de grises, a la cual se le añadió ruido del tipo sal y pimienta, de igual manera e muestra el resultado de usar el filtro de apertura y cierre anteriormente descrito.

Otra manera de usar utilizar el filtro de apertura y cierre, es aplicarlo repetitivamente, variando en cada iteración el tamaño de la estructura de referencia. A este proceso se le conoce como **Alternación del filtro secuencial** (*‘Alternating secuencial filter’*). El efecto de este filtro sobre imágenes a escala de grises permite obtener resultados más suaves que los obtenidos por la simple operación del filtro de apertura y cierre. El algoritmo 9.2 muestra el proceso de implementación de la alternación del filtro secuencial.

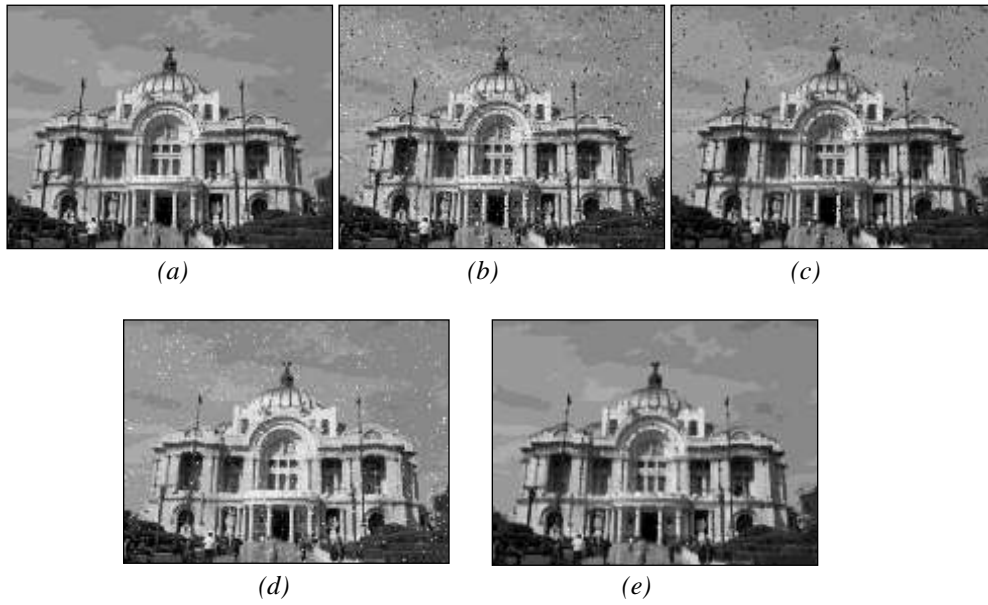
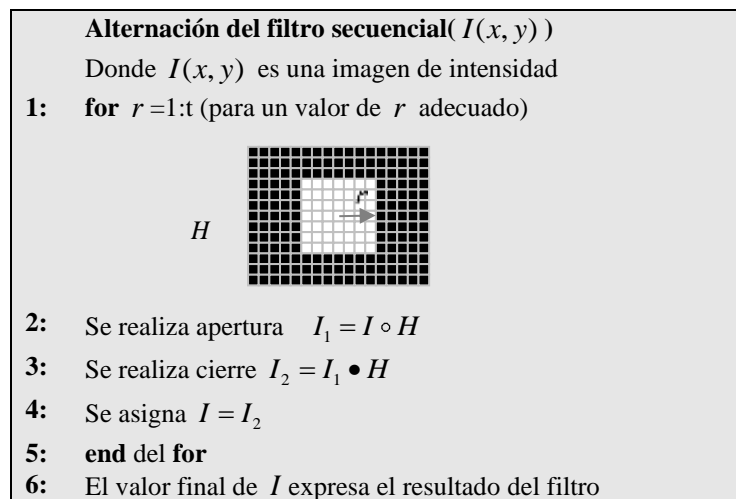


Figura 9.23 Resultados de las operaciones apertura y cierre sobre imágenes a escala de grises. (a) Imagen original, (b) imagen con ruido de sal y pimienta, (c) resultado de la operación apertura sobre (b), (d) resultado de la operación cierre sobre (b) y (e) resultado del filtro de apertura y cierre sobre (b).



Algoritmo 9.2 Algoritmo de implementación de la alternación del filtro secuencial.

La figura 9.24 muestra el efecto de la alternación del filtro secuencial.

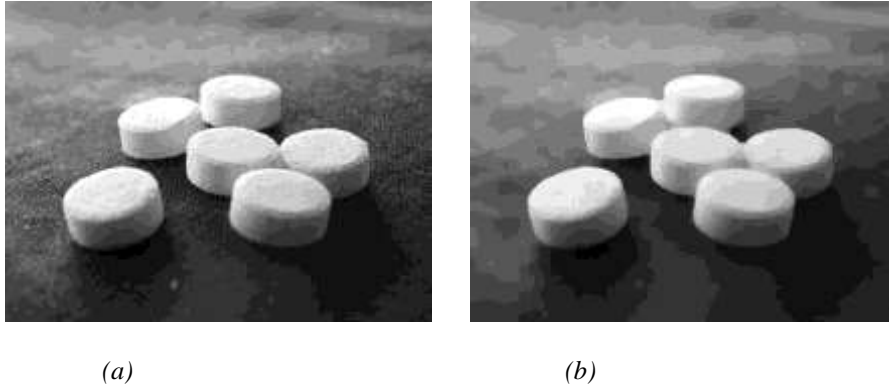


Figura 9.24 Efecto de la alternación del filtro secuencial. (a) Imagen original y (b) el resultado del filtro, obtenido a partir de una repetición de 3 etapas variando  $r$  de 1 a 3.

### 1.5.4 Transformación top-hat y bottom-hat

Esta operación se encuentra definida por la resta de la imagen original a escala de grises, menos la apertura de esa imagen; es decir:

$$I_{TopH} = I - (I \circ H) \quad (9.21)$$

La operación toma su nombre de la forma que presenta la estructura de referencia en esta operación, la cual como muestra la figura 9.25 es similar a la de un sombrero.

1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0

Figura 9.25 Estructura de referencia usada en la transformación top-hat.

Esta operación es útil para el realzado de detalles ante la presencia de sombras. La figura 9.26 muestra el resultado de esta operación en imágenes.

La operación bottom-hat se define como la resta realizada entre el cierre de una imagen y la imagen original, tal que:

$$I_{BottomH} = (I \circ H) - I \quad (9.22)$$

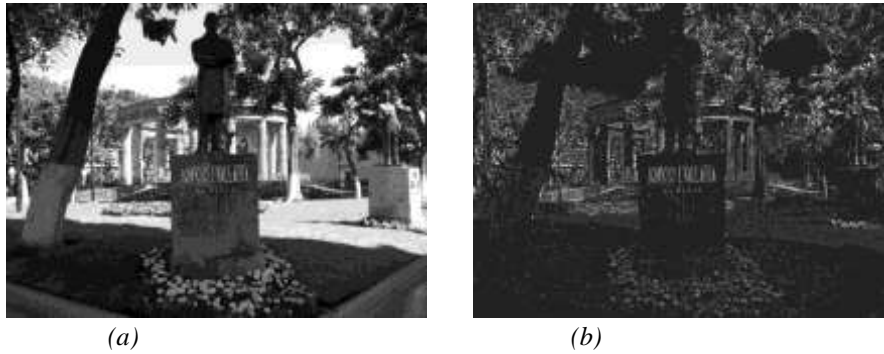


Figura 9.26 Resultado de la transformación top-hat. (a) Imagen original y (b) resultado de la transformación Top-hat.

## 1.6 FUNCIONES DE MATLAB PARA OPERACIONES MORFOLÓGICAS

MatLAB posee un gran número de funciones que permiten implementar la mayoría de las operaciones morfológicas expuestas en secciones anteriores. En esta sección se explican una a una, las funciones que permiten ejecutar las operaciones morfológicas así como algunas funciones auxiliares.

### 1.6.1 Función `strel`

Todas las operaciones morfológicas realizan su función sobre imágenes, utilizando para ello un elemento estructural denominado estructura de referencia. MatLAB dispone de la función `strel` la cual permite construir estructuras de referencia de diferente tamaño y forma. Su sintáxis básica es:

```
esref=strel(forma, parámetros);
```

donde `forma` es una cadena de caracteres que especifica la forma a implementar, de la cual se tienen varias predefinidas. Mientras que `parámetros` es una lista de datos que especifican propiedades de la forma. Por ejemplo, `strel('disk', 3)` regresa una estructura de referencia en forma de disco con una dimensión de  $\pm 3$  píxeles de radio. La tabla 9.1 resume las formas y los parámetros que la función `strel` puede implementar.

En este capítulo fue tratado la propiedad de dividir una estructura de referencia en varias para implementar la operación morfológica de dilatación cuando se involucraba estructuras de referencia de grandes dimensiones. Con la finalidad de acelerar la ejecución de las operaciones al momento de utilizar funciones morfológicas, `strel` construye estructuras de referencia compuestas que permiten en forma asociativa representar una estructura de referencia de grandes dimensiones.

Si se considera el caso anteriormente tratado como ejemplo se tiene:

```
>> componentes=getsequence(esref);
>> whos
  Name              Size              Bytes  Class    Attributes

  componentes       6x1              1870   strel
  esref             1x1              2599   strel
>> esref=strel('disk',3)

esref =

Flat STREL object containing 25 neighbors.
Decomposition: 6 STREL objects containing a total of 14
neighbors

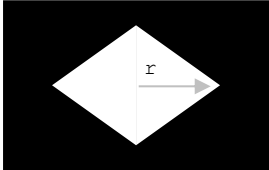
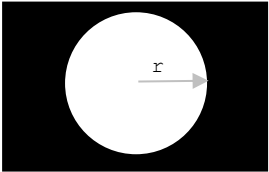
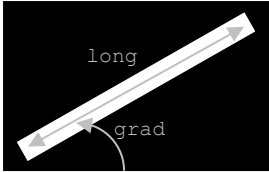
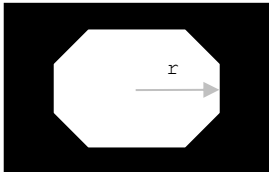
Neighborhood:
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
```

De lo anterior puede notarse como `esref` no es considerada como una matriz normal, sino como una estructura especial llamada objeto `strel`. Este objeto se encuentra compuesto de diferentes elementos. El elemento `Neighborhood`, este elemento es una matriz de unos y ceros que define la estructura de referencia. Otro elemento informa el número de unos contenidos en la estructura que en este caso es de 25. Una parte importante del objeto es la cantidad de elementos en las que puede dividirse la estructura de referencia (en el caso de este ejemplo es 6), de tal forma que si estos elementos se asocian formularían la estructura de referencia completa. La otra información contenida dentro de objeto `strel` es la cantidad de unos presentes en las estructuras en las que se descompuso la estructura de referencia, que en este caso será 14. De lo anterior resulta claro que el criterio para la separación de la estructura de referencia es el número de unos.

La función `getsequence` permite extraer los elementos contenidos en la estructura `strel`. Para el caso de nuestro ejemplo, se tiene:

El comando `whos` arroja como resultado el numero de variables que se encuentran presentes en el ambiente de MatLAB, que en este caso muestra `esref` y `componentes`, los cuales son objetos `strel`. Los 6 elementos con los que se descompuso la estructura pueden ser visualizados indexando el elemento componente, de tal forma que se quiere visualizar el componente número 1 se tendría:

```
>> componentes(1)
ans =
Flat STREL object containing 3 neighbors.
Neighborhood:
    1
    1
    1
```

Sintaxis	Descripción
<code>esref=strel('diamond',r)</code>	<p>Crea una estructura en forma de diamante, donde el valor de r especifica la distancia del extremo de la figura al centro.</p>  <p>The diagram shows a white diamond shape on a black background. A horizontal line with an arrow at the right end extends from the center of the diamond to its rightmost vertex. This line is labeled with the variable 'r'.</p>
<code>esref=strel('disk',r)</code>	<p>Crea una estructura de forma de disco, donde r especifica el radio del mismo.</p>  <p>The diagram shows a white circular disk on a black background. A horizontal line with an arrow at the right end extends from the center of the disk to its right edge. This line is labeled with the variable 'r'.</p>
<code>esref=strel('line',long,grad)</code>	<p>Construye una estructura en forma de línea, donde el parámetro long, representa la longitud de la línea, mientras que grad, define la dirección de la línea.</p>  <p>The diagram shows a white line segment on a black background. The segment is oriented diagonally. A label 'long' is placed above the segment, and a label 'grad' is placed below it, next to a curved arrow indicating the angle of the segment relative to the horizontal.</p>
<code>esref=strel('octagon',r)</code>	<p>Crea una estructura octagonal, donde r especifica la distancia del centro a uno de los lados.</p>  <p>The diagram shows a white octagon on a black background. A horizontal line with an arrow at the right end extends from the center of the octagon to its rightmost side. This line is labeled with the variable 'r'.</p>

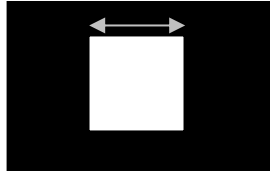
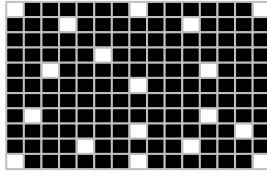
<pre>esref=strel('square',w)</pre>	<p>Crea una estructura cuadrada de ancho definido por w.</p> 
<pre>esref=strel('arbitrary', Matriz)</pre>	<p>Construye una estructura de una forma arbitraria definida por una Matriz cuyos elementos son ceros y unos.</p> 

Tabla 9.1 Formas y estructuras que pueden ser implementadas mediante la función *strel* de MatLAB.

## 1.6.2 Funciones de MatLAB para la dilatación y erosión

La función `imdilate` implementa la operación de dilatación y su sintaxis básica es

```
IR=imdilate(I,H)
```

Donde  $I$ , es la imagen a la cual se le aplica la dilatación usando como estructura de referencia  $H$ .  $IR$  es la imagen dilatada, si  $I$  es una imagen binaria, entonces  $IR$  será también binaria, mientras que si  $I$  es una imagen de intensidad  $IR$  también lo será. Por lo que al utilizar la misma función podemos desempeñar esta operación para imágenes binarias y para imágenes a escala de grises. La estructura de referencia  $H$  puede obtenerse mediante la utilización de la función `strel`.

La función `imerode` implementa la operación de erosión y su sintaxis básica es

```
IR=imerode(I,H)
```

Donde  $I$ , es la imagen a la cual se le aplica la erosión usando como estructura de referencia  $H$ .  $IR$  es la imagen erosionada, si  $I$  es una imagen binaria, entonces  $IR$  será también binaria, mientras que si  $I$  es una imagen de intensidad  $IR$  también lo será. Por lo que al utilizar la misma función podemos desempeñar esta operación para imágenes binarias y para imágenes a escala de grises. La estructura de referencia  $H$  puede obtenerse mediante la utilización de la función `strel`.

Con el objetivo de mostrar la forma de utilizar las funciones de dilatación y erosión se implementara la forma de encontrar el **gradiente morfológico** (sección 9.5.2) de una imagen a escala de grises.



Suponiendo que  $I$  es la imagen a la cual se le quiere extraer el gradiente morfológico, se utilizan los siguientes comandos:

```
>>H=strel('square',3);  
>>I1=imdilate(I,H);  
>>I2=imerode(I,H);  
>>IG=I1-I2;
```

Donde  $H$  es la estructura de referencia que se construye a partir de una forma cuadrada de dimensión 3. Al obtenerse la dilatación y la erosión utilizando la misma imagen y la misma estructura de referencia se realiza una substracción entre la versión dilatada y erosionada de la imagen, encontrando así el gradiente morfológico de  $I$ .

### 1.6.3 Funciones de MatLAB que involucran las operaciones de apertura y cierre.

Las funciones `imopen` e `imclose` implementan las operaciones de apertura y cierre, las sintaxis básicas de estas funciones son:

```
IR=imopen(I,H)  
IR=imclose(I,H)
```

Donde  $I$ , es la imagen a la cual se le aplica la apertura o cierre usando como estructura de referencia  $H$ .  $IR$  es la imagen resultado, si  $I$  es una imagen binaria, entonces  $IR$  será también binaria, mientras que si  $I$  es una imagen de intensidad  $IR$  también lo será. La estructura de referencia  $H$  puede obtenerse mediante la utilización de la función `strel`.

#### La transformación de éxito o fracaso ('Hit-or-miss')

En la transformación de éxito o fracaso (sección 9.4.4) se utiliza una estructura de referencia constituida realmente por dos estructuras  $H = (H_1, H_2)$ , de tal forma que la transformada de éxito o fracaso se define como:

$$I \otimes H = (I \ominus H_1) \cap (I^c \ominus H_2) \quad (9.23)$$

El toolbox de procesamiento de imagen implementa la transformada de éxito o fracaso mediante la función `bwhitmiss`, cuya sintaxis es descrita por:

```
IR=bwhitmiss(I,H1,H2);
```


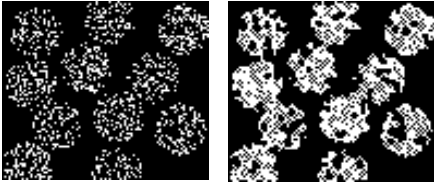

Donde  $IR$  es la imagen obtenida como resultado de esta transformación mientras que  $H_1$  y  $H_2$  son las estructuras de referencia descritas en la ecuación 9.19.

### 1.6.4 La función `bwmorph`

La función `bwmorph` implementa una variedad de operaciones útiles, basadas en combinaciones de dilataciones y erosiones. La sintaxis básica de esta función es:

```
IR= bwmorph(I,operación,n);
```

Donde `IR` es una imagen binaria, `operación` es una cadena de caracteres que especifica la operación que se desea realizar y `n` es un entero positivo que especifica el número de veces que la operación se repite. El termino `n` es opcional y al omitirse significa que la operación se desempeña solo una vez. La Tabla 9.2 describe el conjunto de operaciones que pueden ser llevadas a cabo con la función `bwmorph`. Varias de estas operaciones fueron ya tratadas en este capítulo. La función `bwmorph` puede ser considerada también como una función que implementa rápidamente operaciones morfológicas de gran uso sin necesidad de construir separadamente la estructura de referencia, tal es el caso para la erosión y dilatación donde se realiza la operación utilizando una estructura de referencia de una matriz de unos de 3x3.

Operación	Descripción
<code>bothat</code>	Desempeña la operación bottom-hat, usando una estructura de referencia de tamaño 3x3. Esta función utiliza <code>imbohat</code> para desempeñar la operación. 
<code>bridge</code>	Conecta píxeles los cuales se encuentran separados por huecos de un píxel. 
<code>clean</code>	Remueve píxeles solitarios contenidos en la imagen. 
	Llena los píxeles alrededor de píxeles blancos conectados diagonalmente, para eliminar la conectividad de los píxeles del fondo.

<b>diag</b>	
<b>fill</b>	<p>Llena los huecos presentes en estructuras de objetos presentes en imágenes.</p>
<b>hbreak</b>	<p>Remueve las estructuras H de objetos de la imagen, de tal manera que pueda existir de esta manera un mayor número de objetos.</p>
<b>majority</b>	<p>Coloca el píxel en uno si más de 4 píxeles dentro de una vecindad de 3x3 son unos.</p>
<b>remove</b>	<p>Coloca el píxel en cero si los píxel 4-conectados son unos.</p>
<b>shrink</b>	<p>Si <math>n=\text{inf}</math>, los objetos son reducidos a puntos. Los objetos sin agujeros son reducidos a puntos, mientras que los objetos con agujeros forman anillos. Esta operación mantiene el número de Euler.</p>



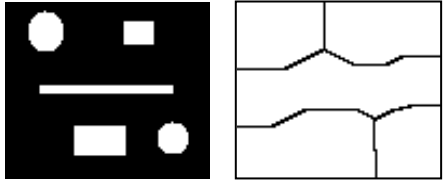

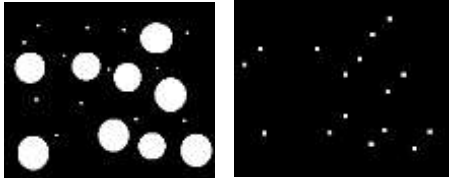
	
<code>skel</code>	<p>Si <math>n=\text{inf}</math> elimina todos los píxeles reduciéndolos a su mínima expresión, líneas. Este proceso es llamado esqueletización.</p> 
<code>thicken</code>	<p>Añade píxeles a los objetos, colocando únicamente píxeles que estén conectados.</p> 
<code>thin</code>	<p>Reduce a los objetos sin agujeros a líneas y con agujeros a anillos.</p> 
<code>tophat</code>	<p>Desempeña la operación bottom-hat, usando una estructura de referencia de tamaño 3x3. Esta función utiliza imbohat para desempeñar la operación.</p> 

Tabla 9.1 Diferentes operaciones que pueden ser implementadas mediante la función **bwmorph** de MatLAB.

### 1.6.5 Etiquetado de componentes convexos

Los componentes convexos son todos los píxeles que tienen valor binario de uno y se encuentran conectados unos con otros en base a algún criterio de vecindad. A cada uno de los objetos encontrados, los cuales forman regiones, se les asigna una etiqueta única que los identifica.

Los algoritmos para el etiquetado se fundamentan principalmente en el criterio de conectividad. Los tipos de conectividad son tres: conectividad-4, conectividad-8 y conectividad diagonal. En la conectividad se dice que un píxel en cuestión (P0) está 4-conectado (figura 9.27(a)) si dicho píxel tiene un uno como vecino en la parte de arriba (P2), abajo (P7), del lado derecho (P5) o bien del lado izquierdo (P4). Un píxel se dice que está 8-conectado (figura 9.27(b)) si el píxel en cuestión (P0) presenta una vecindad con otro uno colocado, además de las posiciones de la vecindad 4-conectado (P2, P4, P5 y P7) de los elementos diagonales (P1, P3, P6 y P8).

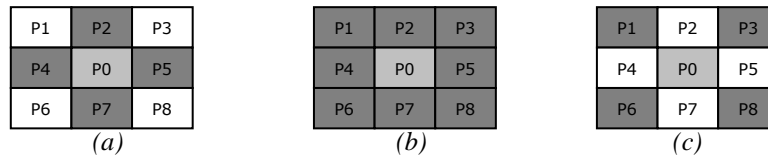


Figura 9.27 Criterios de vecindad entre píxeles. (a) Vecindad 4-conectados, (b) vecindad 8-conectados y (c) diagonal conectados.

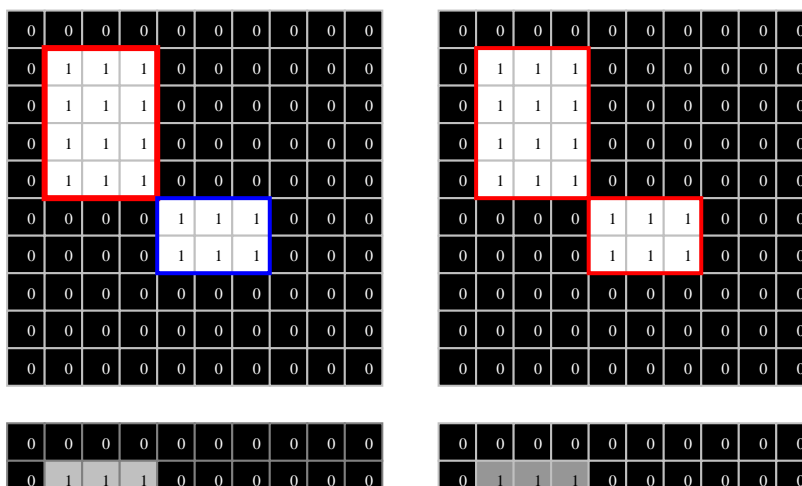
La conectividad diagonal (figura 9.26(c)) se define como la vecindad que presenta un píxel en cuestión con sus vecinos diagonales (P1, P3, P6 y P8).

El criterio de vecindad usado tiene una gran implicación en el resultado de los objetos encontrados en la imagen. En la figura 9.28 se ilustra la diferencia en el etiquetado de objetos dependiendo del criterio de vecindad. La figura 9.28(a) muestra los objetos encontrados cuando se usa la vecindad 4-conectados, mientras que en la figura 9.28(b) se observa el resultado cuando el criterio usado es el 8-conectados. Como puede verse de ambas figuras, resulta claro que cuando se usa el criterio 4-conectados se tendrán más objetos, ya que el criterio de conexión entre objetos será más restrictivo al considerar solo 4 píxeles, mientras que en el caso de 8-conectado la vecindad expresa una mayor flexibilidad, pudiendo de esta manera absorber píxeles y considerar a dos objetos como uno solo.

El toolbox de procesamiento de imágenes de MatLAB posee la función `bwlabel`, la cual etiqueta los elementos con cierta conectividad, encontrados en una imagen. La sintaxis de esta función es:

```
[L, num]=bwlabel(I, con)
```

donde `I` es una imagen binaria sobre la cual se pretende etiquetar los objetos encontrados y con especifica el tipo de criterio de vecindad considerado (4 o 8). `L` es una matriz de los elementos etiquetados encontrados mientras que `num` (este parámetro es opcional) especifica el número de objetos encontrados en la imagen. Si el valor de `con` es omitido la conectividad por defecto es 8-conectados. Los objetos contenidos en `L` se encuentran etiquetados de 1 al número de objeto encontrado.



(a)

(b)

*Figura 9.28 Resultados de etiquetado de objetos al considerar diferentes criterios de vecindad. (a) Resultado al considerar vecindad 4-conectados y (b) resultado al considerar vecindad 8-conectados.*

Si se considera la imagen 9.29(a) como la imagen binaria a la cual se desea encontrar objetos, el resultado de aplicar la función `bwlabel` será una imagen de 3 etiquetas. Para seleccionar cada una de las etiquetas contenidas se puede usar una función lógica que permite obtener la etiqueta en cuestión. Por ejemplo para seleccionar la etiqueta 2 se tiene:

```
L=bwlabel(I,4);  
I2=L==2;
```

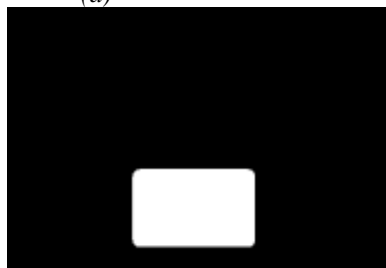
Obteniéndose como resultado el mostrado en la figura 9.29(c).



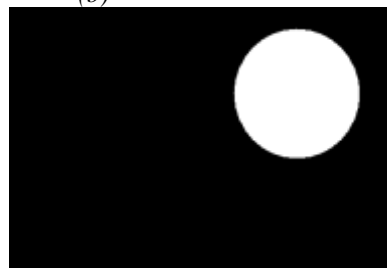
(a)



(b)



(c)



(d)

Figura 9.29 Resultado de la aplicación de la función `bwlabel` para el etiquetado de objetos convexos. (a) Imagen binaria original a la cual se le quieren detectar los objetos presentes, (b) resultado de usar  $L=1$ , (c) resultado de usar  $L=2$  y (d) resultado de usar  $L=3$ .

## 1.7 BLOQUES DE SIMULINK PARA OPERACIONES MORFOLÓGICAS

La librería para el procesamiento de imágenes y video de Simulink posee una serie de bloques que permiten implementar la mayor parte de las operaciones morfológicas tratadas en este capítulo.

### 1.7.1 Bloques de las principales operaciones morfológicas

#### Dilate (Dilatación)

Este bloque de la librería de procesamiento de imágenes y video de Simulink, ubicado en la categoría '*Morphological operations*' (Operaciones morfológicas), permite desempeñar la operación de dilatación sobre una imagen. La figura 9.30 muestra la representación de este bloque.

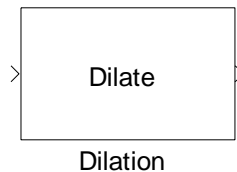


Figura 9.30 Bloque que implementa la operación morfológica de dilatación.

La estructura de referencia usada por esta operación morfológica puede ser configurada de dos maneras diferentes: a) mediante su definición a través de la mascarilla de configuración, o b) por medio de su definición de forma externa, utilizando un puerto de entrada extra. La figura 9.31 muestra la mascarilla de configuración de este bloque.

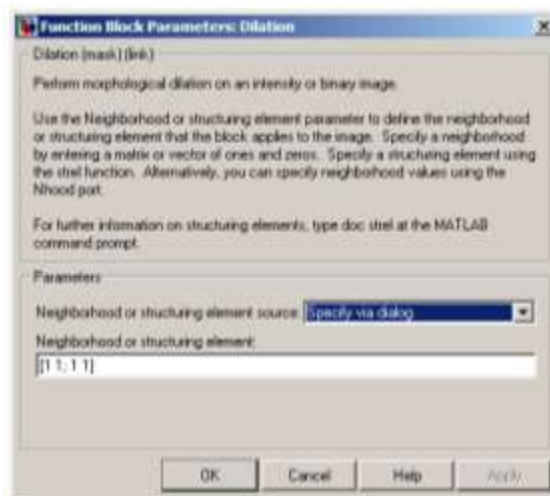


Figura 9.31 Ventanilla de configuración del bloque Dilate.

De la ventana de configuración, se observa que la estructura de referencia puede ser definida mediante la colocación de una matriz de unos y ceros en el campo ‘*Neighborhood or structuring element*’. Esta opción se prefiere cuando la estructura de referencia es compacta. Una forma atractiva de definir una estructura de referencia usando esta misma opción, es utilizar la función **strel** para definir la estructura mediante la digitación de la función, definiendo su forma y tamaño. Sin embargo cuando la estructura de referencia es de tamaño considerable, sería mejor usar un puerto extra donde se acople de manera externa al bloque una matriz que defina a la estructura. Para utilizar esta opción solo hay que seleccionar en el campo ‘*Neighborhood or structuring element source*’ la opción ‘*Input port*’. Si se elige esta opción, el bloque gráficamente mostrará un puerto adicional donde puede acoplarse la estructura de referencia, al usar quizás el bloque ‘*Image From Workspace*’. La figura 9.32 muestra la representación de esta configuración.

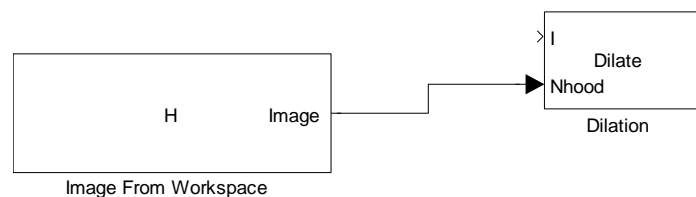


Figura 9.32 Configuración externa de la estructura de referencia.

El bloque detecta el tipo de imagen de que se trata, si la imagen de entrada es binaria el operador se define para este tipo de imagen dando como salida una imagen binaria; sin embargo, si la imagen es a escala de grises, el operador morfológico deberá definirse para este tipo de imagen, dando como salida una imagen de intensidad.

### Erode (Erosión)

Este bloque de la librería de procesamiento de imágenes y video de Simulink, ubicado en la categoría “Morphological operations” (Operaciones morfológicas), permite desempeñar la operación de erosión sobre una imagen. La figura 9.33 muestra la representación de este bloque.

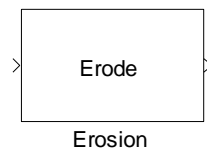


Figura 9.33 Bloque que implementa la operación morfológica de dilatación.

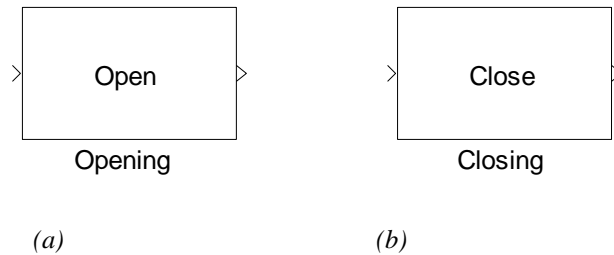
La forma de definir la estructura de referencia corresponde a las utilizadas para el caso de la operación de dilatación.

### Opening y Closing (apertura y cierre)

Estos bloques de la librería de procesamiento de imágenes y video de Simulink, ubicados en la categoría ‘*Morphological operations*’ (Operaciones morfológicas), permiten desempeñar las



operaciones de apertura y cierre sobre una imagen. La figura 9.34 muestra las representaciones de estos bloques.



*Figura 9.34 Bloques que implementan la operación morfológica de (a) apertura y (b) cerradura.*

Para definir la estructura de referencia, se tienen también dos diferentes opciones, al igual que en el caso de las operaciones de dilatación y erosión. Es decir, todas las características anteriormente vistas para los bloques erosión y dilatación, son completamente transportables para el caso de las operaciones apertura y cerradura.

Estos bloques, al igual que los anteriores, identifican el tipo de imagen que se acopla a su entrada, definiendo el operador morfológico a utilizar según si la imagen es binaria o de intensidad.

### **1.7.2 Ejemplos de la utilización de los bloques de Simulink de las principales operaciones morfológicas.**

Para ilustrar la forma en como se utilizan los bloques de dilatación, erosión, apertura y cierre se realizarán 3 aplicaciones sencillas: Detección de bordes por erosión, gradiente morfológico y eliminación de artefactos en imágenes de bordes.

#### **Detección de bordes por erosión**

En este ejemplo se utilizarán los bloques de Simulink para implementar el algoritmo de detección de bordes por erosión visto en la sección 9.3. La idea de la implementación es detectar los bordes de una imagen binaria, obtenida por la umbralización de la imagen proveniente de una Webcam. La configuración del programa se encuentra especificada en la figura 9.35. El flujo del programa de esta aplicación, empieza con la imagen entregada por dicha Webcam. Esta imagen se convierte a escala de grises. A la imagen de intensidad se le aplica un umbral con el objetivo de producir una imagen binaria.

El proceso para la detección, el cual es descrito en el algoritmo 9.1, comienza con aplicar la operación morfológica de erosión sobre la imagen, utilizando como estructura de referencia alguna de las definidas en la figura 9.10, las cuales corresponden a relaciones de vecindad. El objetivo de aplicar la erosión es el de eliminar de la imagen original la capa exterior (borde), de los objetos dentro de la imagen. Al considerar lo anterior, el borde de los objetos de la imagen puede encontrarse si de la imagen erosionada se obtiene la inversa, para tener con ello una imagen donde los objetos tendrán un valor de cero, mientras que el fondo de la imagen más los bordes de los objetos estarán en uno. Por lo que si se hace una operación AND entre la imagen original y la inversa de la versión erosionada de la imagen, se obtendrán los bordes de los objetos, ya que los únicos píxeles comunes entre ambas imágenes son los bordes.

Para la creación de este programa se usan los bloques de Simulink, aceptando su configuración original, por lo que basta con posicionar cada uno de los bloques presentados en la figura 9.35 sobre el modelo. La figura 9.36 muestra el resultado de ejecutar el programa de Simulink mostrado en la figura 9.35.

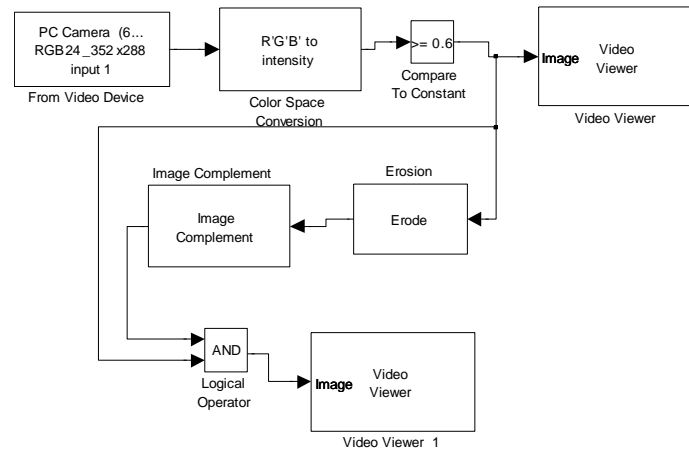


Figura 9.35 Programa en Simulink que implementa la detección de bordes por erosión.

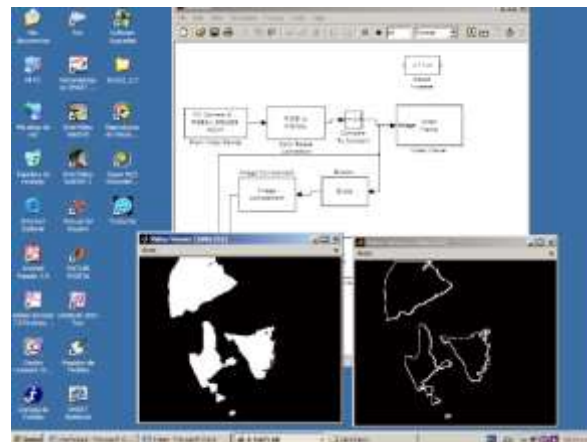


Figura 9.36 Resultado de ejecutar el programa mostrado en la figura 9.35.

## Gradiente Morfológico

En este ejemplo se utilizarán los bloques de Simulink para implementar el gradiente morfológico visto en la sección 9.5.2. La idea de la implementación es calcular el gradiente morfológico de una imagen a escala de grises proveniente de una Webcam. La configuración del programa se encuentra especificada en la figura 9.37. El flujo del programa de esta aplicación empieza con la imagen entregada por tal Webcam. A partir de esta imagen es realizada la conversión a escala de grises.

El cálculo del gradiente se realiza mediante la substracción que se lleva a cabo entre la versión dilatada de la imagen, menos la versión erosionada, tal que:

$$\partial M = (I \oplus H) - (I \ominus H) \quad (9.24)$$

Para la creación de este programa se usan los bloques de Simulink, aceptando su configuración original, por lo que para su creación basta con posicionar cada uno de los bloques presentados en la figura 9.37 sobre el modelo. El bloque de ganancia que multiplica por 4 al gradiente morfológico se hace necesario para realzar la visualización del mismo. La figura 9.40 muestra el resultado de haber ejecutado el programa mostrado en la figura 9.38.

### Eliminación de artefactos en imágenes de bordes.

Independientemente del algoritmo usado para la detección de bordes en una imagen, el resultado obtenido contiene una serie de artefactos, producto o bien del ruido presente o de la incertidumbre del algoritmo en la localización del borde. Sea cual fuese el caso, estos artefactos pueden generar problemas al momento de realizar operaciones posteriores sobre la imagen, tales como clasificación o reconocimiento.

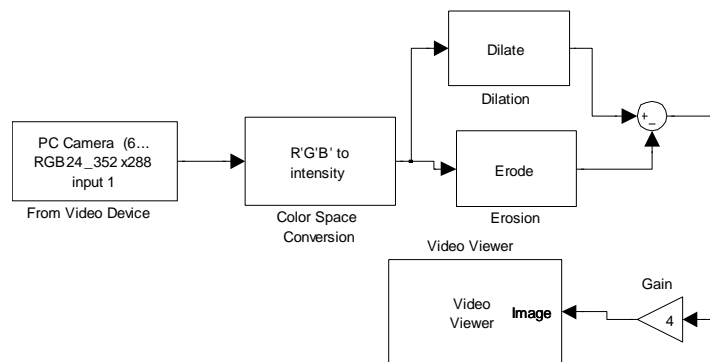


Figura 9.37 Programa en Simulink que implementa el gradiente morfológico de una imagen a escala de grises.

La forma más efectiva de eliminar estos artefactos es, primero erosionar la imagen y después dilatarla. La erosión eliminaría los artefactos, sin embargo adelgazaría algunos bordes que pudieran ser importantes; mientras que al dilatar la imagen los bordes adelgazados serán reconstruidos. Como fue visto en la sección 9.4 la operación que desempeña una erosión seguida de una dilatación es la operación ‘Opening’ (Cierre).

En este ejemplo se utilizaran los bloques de Simulink para la eliminación de los artefactos en imágenes de bordes. La idea de la implementación es aplicar la operación de Cierre a una imagen donde previamente se han calculado los bordes, a través de un algoritmo adecuado, como podría ser el Sobel (Capítulo 6). La configuración del programa se encuentra especificada en la figura 9.38. El flujo del programa de esta aplicación empieza con la imagen entregada por una Webcam. A partir de esta imagen se extraen los bordes mediante el bloque ‘Edge Detection’.

Después, a la imagen se le realiza la operación de apertura, utilizando como estructura de referencia la de un disco de radio 3, generada por la función `strel('disk', 3)`. Dicha función debe definirse dentro de la ventana de configuración del bloque ‘Opening’.

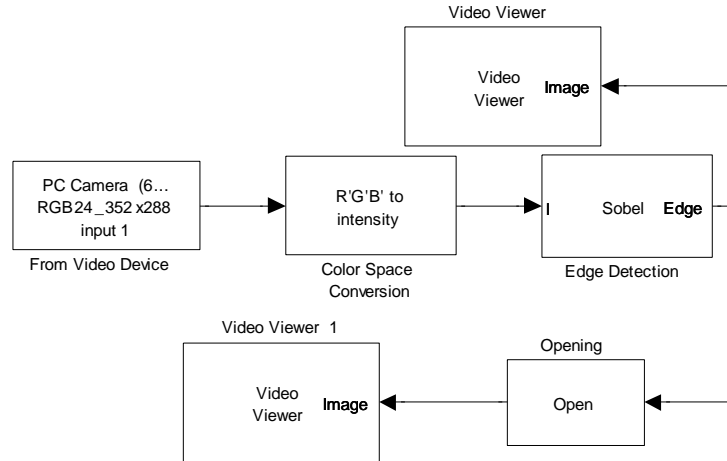


Figura 9.38 Programa en Simulink que implementa la eliminación de artefactos de una imagen de bordes.

### 1.7.3 Bloques de las operaciones auxiliares morfológicas.

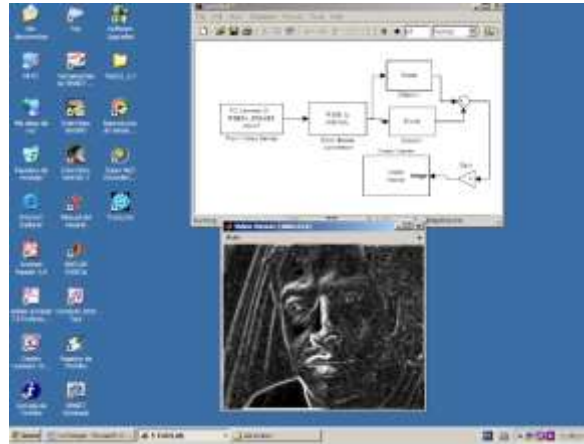
#### Label (Bloque de etiquetación de objetos convexos)

Este bloque de la librería de procesamiento de imágenes y video de Simulink, ubicado en la categoría '*Morphological operations*' (Operaciones morfológicas), permite etiquetar el número de objetos que se encuentran en una imagen, cuyos píxeles presentan alguna forma de conexión (véase subsección 9.6.5). La figura 9.39 muestra la representación de este bloque.



Figura 9.39 Bloque que implementa el etiquetado de objetos contenidos en una imagen binaria.

El bloque recibe como entrada una imagen binaria **BW** y regresa como salida una matriz **Label** de datos etiquetados (véase subsección 9.6.5), donde se identifica a cada objeto contenido en la imagen binaria cuyos píxeles forman una nube monolítica de datos que se agrupan en base a un criterio de vecindad. El criterio de vecindad empleado para la localización de los objetos es definido desde su ventana de configuración. Otra variable que regresa el bloque es **Count** la cual indica el número de objetos identificados de la imagen **BW**.



*Figura 9.40 Resultado de ejecutar el programa mostrado en la figura 9.37.*

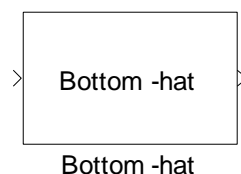
Debido a que el bloque **Label** etiqueta a cada objeto que encuentra en la imagen utilizando una sucesión numérica consecutiva y creciente, la matriz **Label** contendrá solo datos que van de cero (el fondo) hasta  $n$ , siendo  $n$  el numero de objetos encontrados. Si se considera a **Label** como una imagen a escala de grises, el valor de niveles de intensidad estará integrado por  $n+1$ . En estas condiciones, la representación de **Label** prácticamente no sería visualizable mediante la utilización del bloque *Video Viewer* (método estándar utilizado para el despliegue de video en Simulink). Por esta razón, resulta necesario para la visualización de esta matriz utilizar el bloque '*Contrast Adjustment*', o ajuste de contraste (visto en el capítulo 4, sección 4.7.4).

La imagen **Label** obtenida del bloque **Label** contiene pocos niveles de intensidad (dependiendo del número de objetos) y ya que sus niveles de intensidad se encuentran enumerados consecutivamente, estos valores están cargados a los niveles de intensidad oscuros. Estas características evitan que la imagen **Label** sea fácilmente visible.

El bloque '*Contrast Adjustment*' adapta los niveles de intensidad de la imagen **Label**, distribuyéndolos uniformemente en todo el ancho permisible del tipo de dato utilizado para representar la imagen, lo cual permite visualizar la imagen perfectamente asignándole un nivel de intensidad visiblemente detectable para cada objeto etiquetado por el bloque **Label**.

### **Bottom-hat**

Este bloque de la librería de procesamiento de imágenes y video de Simulink, ubicado en la categoría '*Morphological operations*' (Operaciones morfológicas), permite implementar la operación Bottom-hat (véase subsección 9.5.4). La figura 9.41 muestra la representación de este bloque.



*Figura 9.41 Bloque que implementa la operación Bottom-hat de una imagen binaria.*

La estructura de referencia usada por esta operación morfológica puede ser configurada de dos maneras diferentes: a) mediante su definición a través de la mascarilla de configuración, o b) por medio de su definición de forma externa, utilizando un puerto de entrada extra.

### 1.7.4 Ejemplos de la utilización de los bloques de Simulink de las operaciones morfológicas auxiliares.

Para ilustrar la forma en como se utilizan este tipo de bloques se realizara una aplicación sencilla: Visualización de los objetos detectados de una imagen binaria.

En este ejemplo se utilizaran los bloques de Simulink para implementar la visualización de los objetos detectados de una imagen binaria. La idea de la implementación es detectar los objetos de una imagen binaria, obtenida por la umbralización de la imagen proveniente de una Webcam. La configuración del programa se encuentra especificada en la figura 9.42. El flujo del programa de esta aplicación empieza con la imagen entregada por la mencionada Webcam. A partir de esta imagen se convierte a escala de grises. A la imagen de intensidad se le aplica un umbral con el objetivo de producir una imagen binaria.

En el proceso para la detección, se utiliza el bloque **Label**, anteriormente explicado. El bloque permite configurar el criterio de vecindad usado para considerar la concatenación de píxeles, en el caso de este ejemplo dicho criterio se configuro a vecindad-4.

Como ya fue explicado la matriz **Label** obtenida no puede ser visualizada directamente, por lo que previo a la visualización se colocara el bloque 'Contrast Adjustment', el cual permite ajustar los niveles de intensidad presentes (los cuales representan a los objetos binarios conectados) para mejorar su representación. La figura 9.43 muestra el resultado de haber ejecutado el programa mostrado en la figura 9.42.

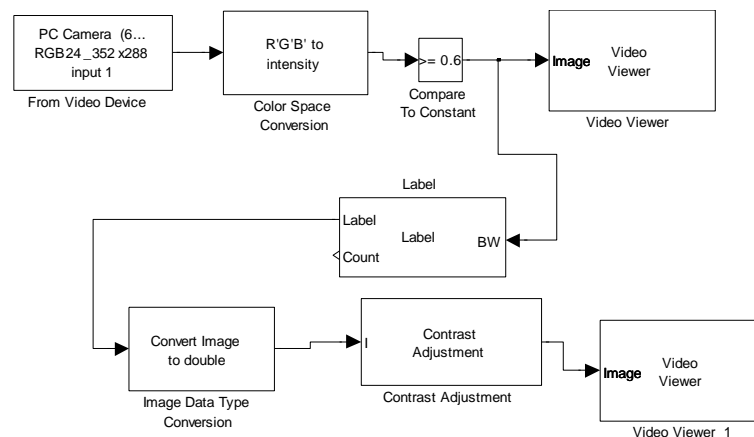


Figura 9.42 Programa en Simulink que implementa la visualización de los objetos conectados encontrados en una imagen binaria.

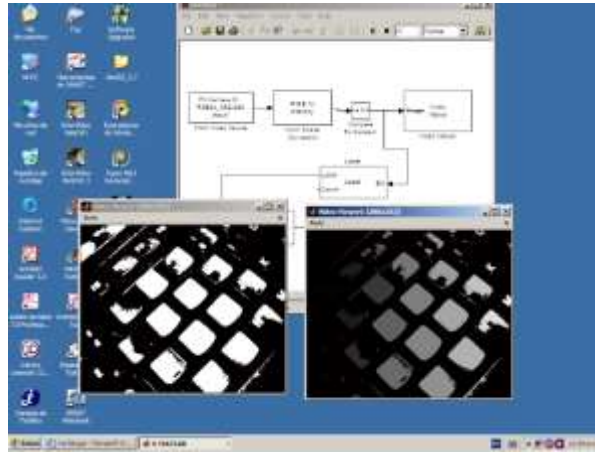


Figura 9.43 Resultado de ejecutar el programa mostrado en la figura 9.41.