Apéndice C

Fundamentos de Morfología Matemática

C.1 INTRODUCCIÓN

La morfología matemática (MM) es desarrollada a partir de los años 60 por los trabajos de G. Matheron y J. Serra, siendo una técnica aplicable a muchos problemas de tratamiento de señales y entre éstas las imágenes.

En este apéndice vamos a describir las herramientas morfológicas básicas, algunas de las cuales se han utilizado en el proyecto. Estas notas se basan en los apuntes de los cursos Albiol et. al. [8] y Serra [28], así como en el capítulo 4 de la tesis de J. M. Mossi [21]. Para el lector que quiera profundizar sobre morfología matemática encontrará un análisis exhaustivo en las obras de J. Serra [26] [27] y de G. Matheron [17] [18].

C.1.1 Conjuntos y funciones en MM

Los operadores morfológicos están en su fundamentación teórica íntimamente ligados a los conjuntos y funciones.

En la forma más habitual de la MM aplicada al tratamiento de imágenes, se definen éstas sobre un espacio euclideo discrito $\mathcal{Z}^2$ y pueden ser binarias o de niveles de gris, según las siguientes definiciones:

Imagen binaria Una imagen binaria I es una aplicación de un subconjunto $\mathcal{D}_I$ de $\mathcal{Z}^2$, llamado dominio de definición de I, en el par $\{0,1\}$:

$$I \,:\, \mathcal{D}_I \subset \mathcal{Z}^2 \longrightarrow \{0,1\}$$

$$p \longrightarrow I(p)$$

Cada elemento p, del subconjunto $\mathcal{D}_I$ donde está definida la imagen se llama pixel. Cuando se trata de imágenes binarias, generalmente van asociadas a escenas donde hay un fondo y uno o varios objetos del mismo tipo, es decir dos clases. Consecuentemente a cada pixel de la imagen se le asigna uno de los dos valores posibles, $\{0, 1\}$, según pertenezca a una clase u otra. El enfoque de la teoría de conjuntos de la MM se aplica directamente a las imágenes binarias sin más que considerar la imagen binaria I como el conjunto de pixels de valor 1, es decir $\{p \in \mathcal{D}_I,\ I(p) = 1\}$, que a su vez es un subconjunto del espacio $\mathcal{D}_I$ en el que está definida. La relación de orden es la inclusión.

Imagen de niveles de gris Una imagen de niveles de gris I es una aplicación de un subconjunto $\mathcal{D}_I$ de $\mathcal{Z}^2$, llamado dominio de definición de I en $\mathcal{Z}$:

$$I \,:\, \mathcal{D}_I \subset \mathcal{Z}^2 \longrightarrow \mathcal{Z}$$

$$p \longrightarrow I(p)$$

Así como a las imágenes binarias se les puede aplicar la teoría de conjuntos, a las imágenes de niveles de grises se aplica la teoría de funciones. Las dos definiciones anteriores se pueden hacer en el espacio de los números reales $I \,:\, \mathcal{D}_I \subset \mathcal{R}^2 \longrightarrow \mathcal{R}$ pero las imágenes que se usan en la práctica están definidas sobre un dominio finito discreto y generalmente son rectangulares. De cara a la utilización que se hace de la MM en nuestro proyecto, si es interesante considerar la definición en $\mathcal{R}$, y además resulta útil interpretar las dos dimensiones del dominio de definición de la imagen junto con la dimensión del nivel de gris que puede tomar cada pixel paar formar un relieve tridimensional donde la altura de cada punto es proporcional a su nivel de gris. En ocasiones este relieve tridimensional conviente descomponerlo en un conjunto de cortes de nivel con planos horizontales para formar imágenes binarias con fondo y objeto. Esta descomposición en imágenes binarias sirve como nexo para poder utilizar también la teoría de conjuntos sobre las imágenes de niveles de gris. En la descomposición cada imagen binaria corresponde al resultado de la función umbral a un determinado nivel de gris t, definida como:

$$U_t(f) = \begin{cases} 1 & \text{si } f(x) \geq t \\ 0 & \text{si } f(x) < t \end{cases}$$

donde f es la imagen, x representa los elementos que pertenecen al dominio donde está definida la imagen y t representa un nivel de gris. Alternativamente la imagen binaria se puede definir únicamente como el conjunto de pixels de valor 1, así pues la función umbral quedará:

$$U_t(f) = \{x | f(x) \geq t\}$$

Todo lo que se acaba de comentar para imágenes de niveles de gris, y como particularización, permite la aplicación de la MM a señales unidimensionales, $g(n)$, definidas en un dominio $\mathcal{D}_g$:

$$g\ :\ \mathcal{D}_g \subset \mathcal{N} \longrightarrow \mathcal{R}$$

$$n \longrightarrow g(n)$$

donde n es un número natural que corresponde al indice, y pertenece al subconjunto ordenado D_g. Con este tipo de funciones hemos trabajado en nuestro proyecto.

C.1.2 Componente conexa

En muchas ocasiones una imagen representa la proyección bidimensional de una escena tridimensional donde hay uno o varios objetos sobre un fondo. Por simplicidad tomemos el ejemplo de un solo objeto sobre un fondo. Esta situación queda reflejada en la imagen como que un conjunto de pixels "pertenecen" al objeto y el resto pertenecen al fondo. La expresión coloquial de "pertenecer al objeto", es un concepto muy importante y formalmente corresponde a la definición de componente conexa que se definirá a continuación. Para ello es conveniente definir primer lo que es un camino.

Camino Un camino C, de longitud $l(C) = n$, entre dos pixels p y q en la celosía S es una $(n+1)$ tupla $(p_0, p_1, \ldots, p_n)$ de pixels tal que:

$$p_0 = p\ y\ p_n = q$$

$$\forall i \in [1, n],\ p_i \in \eta_{p_{i-1}}$$

Las imágenes que se acaban de definir están compuestas por elementos o pixels dispuestos de forma no arbitraria. La disposición de los mismos obedece normalmente a cierto patrón regular a lo largo del dominio bidimensional. Además, esta disposición geométrica de los pixels junto con la relación que les unen forman la celosía S, mencionada en la defínción de camino.

Componente conexa Sea A un conjunto de pixels incluido en el dominio $\mathcal{D}_I$ donde está definida la imagen binaria I y x un pixel de A. La componente conexa de A que contiene a x, $C_x(A)$ es la unión de los caminos con origen x incluidos en A.

Se dirá también que dos pixels p y q so conexos en A si existe un camino con extremos en p y q que esté incluido en A.

Intuitivamente una componente conexa es un conjunto de pixels donde todos ellos mantiene una relación de vecindad con al menos uno de los restantes pixels del conjunto.

C.2 TRANSFORMACIONES MORFOLÓGICAS BÁSICAS

Antes de pasar a las transformaciones morfológicas, aplicadas sobre $\mathcal{P}(\mathcal{R}^2)$, y definidas como,

$$\mathcal{T} \,:\, \mathcal{P}(\mathcal{R}^2) \longrightarrow \mathcal{P}(\mathcal{R}^2)$$

es útil definir algunas propiedades que suelen cumplir y que son:

- Una transformación es **extensiva** si y sólo si su salida es siempre mayor que su entrada. Para el caso binario,

  $$\forall X \in \mathcal{P}(\mathcal{R}^2),\; X \subseteq \mathcal{T}(X)$$

  y para el caso de niveles de gris,

  $$\forall f \in \mathcal{F},\; f \leq \mathcal{T}(f)$$

  de la misma forma es **antiextensiva** si y sólo si su salida es siempre menor que su entrada

  $$\forall X \in \mathcal{P}(\mathcal{R}^2),\; X \supseteq \mathcal{T}(X)$$

  o

  $$\forall f \in \mathcal{F},\; f \geq \mathcal{T}(f)$$

- La transformacion $\mathcal{T}$ será **creciente** si para dos entradas ordenadas sus salidas siguen estando ordenadas, es decir:

  $$\forall X, Y \in \mathcal{P}(\mathcal{R}^2),\; X \subseteq Y \Rightarrow \mathcal{T}(X) \subseteq \mathcal{T}(Y)$$

  o

  $$\forall f, g \in \mathcal{F},\; f \leq g \Rightarrow \mathcal{T}(f) \leq \mathcal{T}(g)$$

  de la misma manera es **decreciente** en caso contrario:

  $$\forall X, Y \in \mathcal{P}(\mathcal{R}^2),\; X \supseteq Y \Rightarrow \mathcal{T}(X) \subseteq \mathcal{T}(Y)$$

  o

  $$\forall f, g \in \mathcal{F},\; f \geq g \Rightarrow \mathcal{T}(f) \leq \mathcal{T}(g)$$

- Respecto a la siguiente propiedad, una transformación $\mathcal{T}$ será **idempotente** si el resultado de aplicarla concatenadamente varias veces es el mismo que al aplicarla una sola vez,

  $$\forall X \in \mathcal{P}(\mathcal{R}^2),\; \mathcal{T}(\mathcal{T}(X)) = \mathcal{T}(X)$$

  o

  $$\forall f \in \mathcal{F},\; \mathcal{T}(\mathcal{T}(f)) = \mathcal{T}(f)$$

- En imágenes binarias se define el complementario.

  Complementario Sea X el conjunto de pixels de la imagen que tienen valor 1, $\mathcal{D}_I$ el dominio donde está definida la imagen y E el conjunto que abarca la totalidad de elementos del dominio de la imagen, el complementario de X denotado como X^c es tal que:

  $$X \cap X^c = \emptyset \;\; y \;\; X \cup X^c = E$$

  En el caso de funciones el complementario de $f(x)$ se corresponde con la inversión: $-f(x)$; si se está utilizando un conjunto de valores positivos, por ejemplo entre 0 y G la inversión se define como $G - f(x)$.

- Finalmente, dos transformaciones $\mathcal{T}_1$ y $\mathcal{T}_2$, serán **duales** si y sólo si el complementario del resultado de $\mathcal{T}_1$ sobre la entrada es el mismo que el resultado de aplicar $\mathcal{T}_2$ sobre el complementario de la entrada:

  $$\forall X \in \mathcal{P}(\mathcal{R}^2),\; \mathcal{T}_1(X)^c = \mathcal{T}_2(X^c)$$

  o

  $$\forall f \in \mathcal{F},\; -\mathcal{T}_1(f) = \mathcal{T}_2(-f)$$

C.2.1 Erosión y Dilatación

Las dos transformaciones más comunes de la morfología matemática, y que son la base de muchas otras transformaciones morfológicas son la **erosión** y la **dilatación**. A continuación se presentan las definiciones con la notación que se utilizará en adelante. Consideraremos en $\mathcal{R}^2$ un punto arbitrario O y llamémosle *origen*, y sea $B \in \mathcal{P}(\mathcal{R}^2)$ un conjunto, denotaremos como el conjunto simétrico traspuesto $\breve{B}$ respecto al origen O, como:

$$\breve{B} = \{-x,\ x \in B\}$$

y B_a a la traslación de B por el vector a, con $a \in \mathcal{R}^2$:

$$B_a = \{x + a,\ x \in B\}$$

consideremos también un conjunto $X \in \mathcal{R}^2$

Dilatación La dilatación de X por B es el conjunto de los x de $\mathcal{R}^2$ tal que la intersección de X y B_x no es el conjunto vacío. Denotaremos esta operación con $\delta_B(X)$:

$$\delta_B(X) = \{x \in \mathcal{R}^2, B_x \cap X \neq \emptyset\}$$

Erosión De la misma manera la erosión de X por B es el conjunto de los x de $\mathcal{R}^2$ tal que B_x esta totalmente contenido en X. Denotaremos esta operación con $\epsilon_B(X)$:

$$\epsilon_B(X) = \{x \in \mathcal{R}^2, B_x \subset X\}$$

El efecto que la dilatación produce es un agrandamiento de los objetos de la imagen, mientras que la erosión los reduce, llegando a eliminarlos si el elemento estructurante no cabe en ellos.

A modo de resumen, se podría decir que los efectos producidos por la dilatación y la erosión binaria son:

Dilatación

- Hacer más gruesas las figuras.
- Si lo objetos están próximos puede unirlos.
- Reduce/elimina los agujeros.
- Los entrantes estrechos son rellenados.

Erosión

- Hacer más finas las figuras.
- Si los objetos son pequeños pueden desaparecer.

- Si los objetos tienen itsmos estrechos pueden partirse.
- Los salientes estrechos son eliminados.
- Si se aplica reiteradamente elimina todos los pixels a 1 de la imagen.

En el caso de funciones o imágenes de niveles de gris, si B es un elemento estructurante plano, la dilatación de la función f en el punto x es el valor máximo de la función f dentro de la ventana de observación definida por B desplazado de manera que el origen de B esté centrado en x:

$$\delta_f(x) = max\{x_k, k \in B\}$$

donde x_k es el valor que toma la imagen en el punto $x + k$, es decir $f(x + k)$.

Igualmente, la erosión se define como:

$$\epsilon_f(x) = min\{x_k, k \in B\}$$

Los efectos producidos por la dilatación y la erosión de imágenes de grises son:

Dilatación

- Equivale a un máximo deslizante de ancho el tamaño del elemento estructurante.
- Elimina los picos negativos (zonas oscuras) más estrechos que el elemento estructurante.
- Estrecha las zonas oscuras más grandes que el EE.
- Ensancha los picos positivos (zonas claras).
- Siempre está por encima de la señal original.

Erosión

- Equivale a un mínimo deslizante de ancho el tamaño del elemento estructurante.
- Elimina los picos positivos más estrechos que el elemento estructurante, y estrecha los más anchos.
- Siempre está por debajo de la señal original.

Las propiedades más importantes de la dilatación y la erosión son que ambas transformaciones son crecientes, y que si el origen del elemento estructurante está contenido en él mismo, entonces, son extensivas y anti-extensivas respectivamente. Ambas funciones son además duales, de manera que:

$$\delta_B(X)^c = \epsilon_B(X^c)$$

Tanto la erosión como la dilatación no son idempotentes. Aplicar sucesivas veces la erosión por ejemplo a imágenes binarias va reduciendo iterativamente la cantidad de pixels pertenecientes a los objetos de la imagen.

$$\delta_B(X) \neq \delta_B(\delta_B(X)), \quad \epsilon_B(X) \neq \epsilon_B(\epsilon_B(X))$$

Algoritmo rápido de erosión de imágenes de grises

Cuando el EE es lineal, es posible aplicar un algoritmo muy eficiente para el cálculo de la erosión (y dilatación) de una imagen de grises (o de una función, como en nuestro proyecto).

Se supondrá que el EE es un segmento horizontal de longitud L. Se debe recorrer la imagen por filas y de izquierda a derecha.

Para calcular un valor de la erosión, debo calcular el mínimo de un conjunto de números:

- $(L-1)$ intervinieron en el cálculo del mínimo anterior.
- 1 pixel nuevo $\mathcal{N}$.
- 1 pixel que ya no se debe tener en cuenta $\mathcal{V}$.

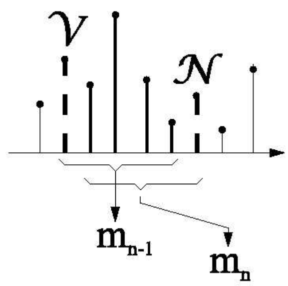


Figura C.1: Esquema del algoritmo rápido de erosión de grises.

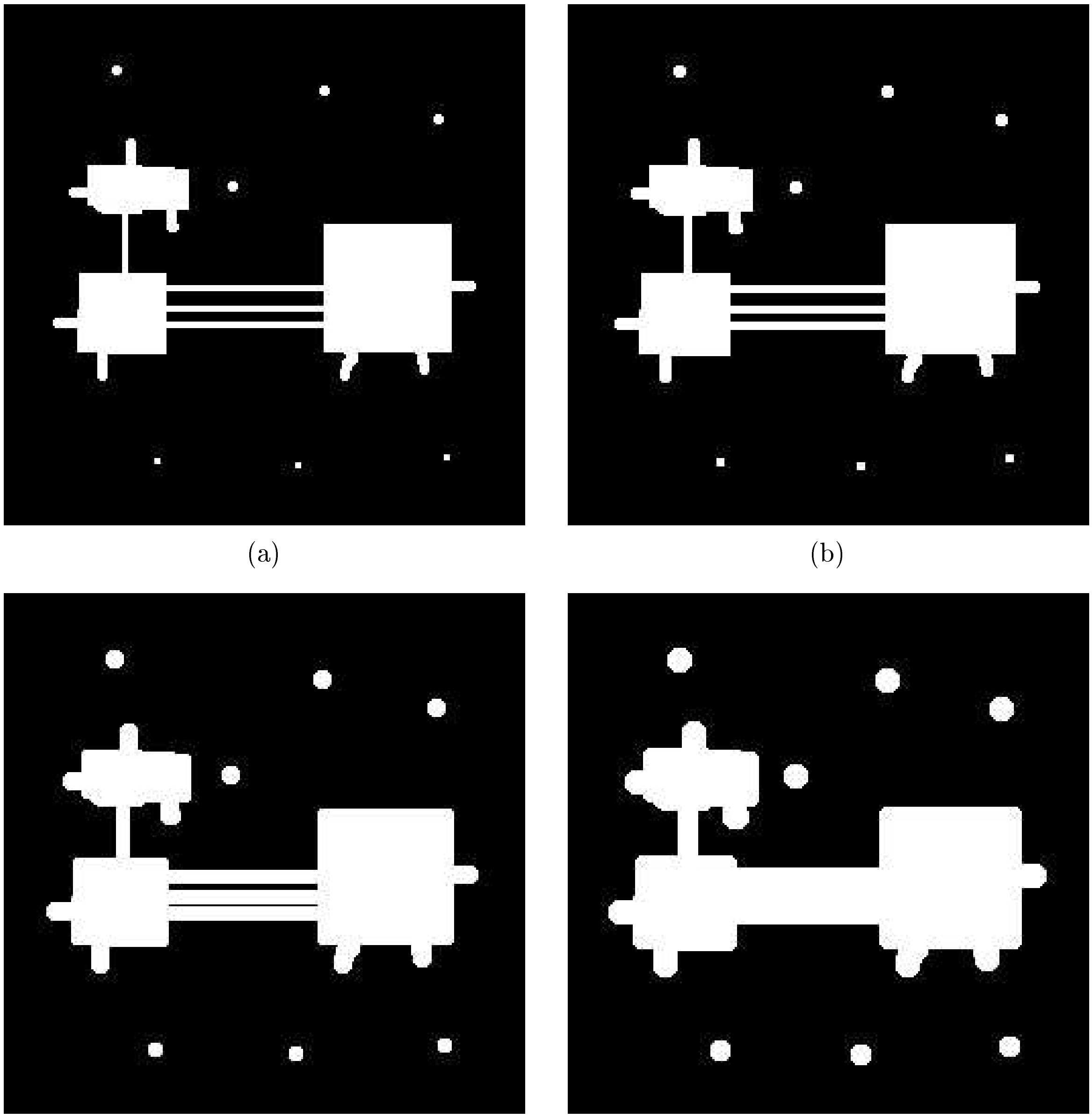
El algoritmo consiste en:

- Si $\mathcal{N} \leq m_{n-1}$ entonces $m_n = \mathcal{N}$.
- En caso contrario, si $\mathcal{V} > m_{n-1}$ entonces $m_n = m_{n-1}$.
- En caso contrario, es necesario buscar el mínimo (el mínimo estaba entre los $L-1$ comunes).

Con este algoritmo el número medio de comprobaciones es:

$$N_{comp}(L) = 1\frac{1}{L+1} + \frac{L}{L+1}(2\frac{L-1}{L} + (L-1)\frac{1}{L})$$

$$\lim_{L\to\infty} N_{comp}(L) = 3$$



(c) (d)

Figura C.2: Efecto de la Dilatación sobre una imagen binaria: (a)Imagen original. (b)Dilatación con EE circular de tamaño 3. (c)Dilatación con EE circular de tamaño 5. (d)Dilatación con EE circular de tamaño 7.

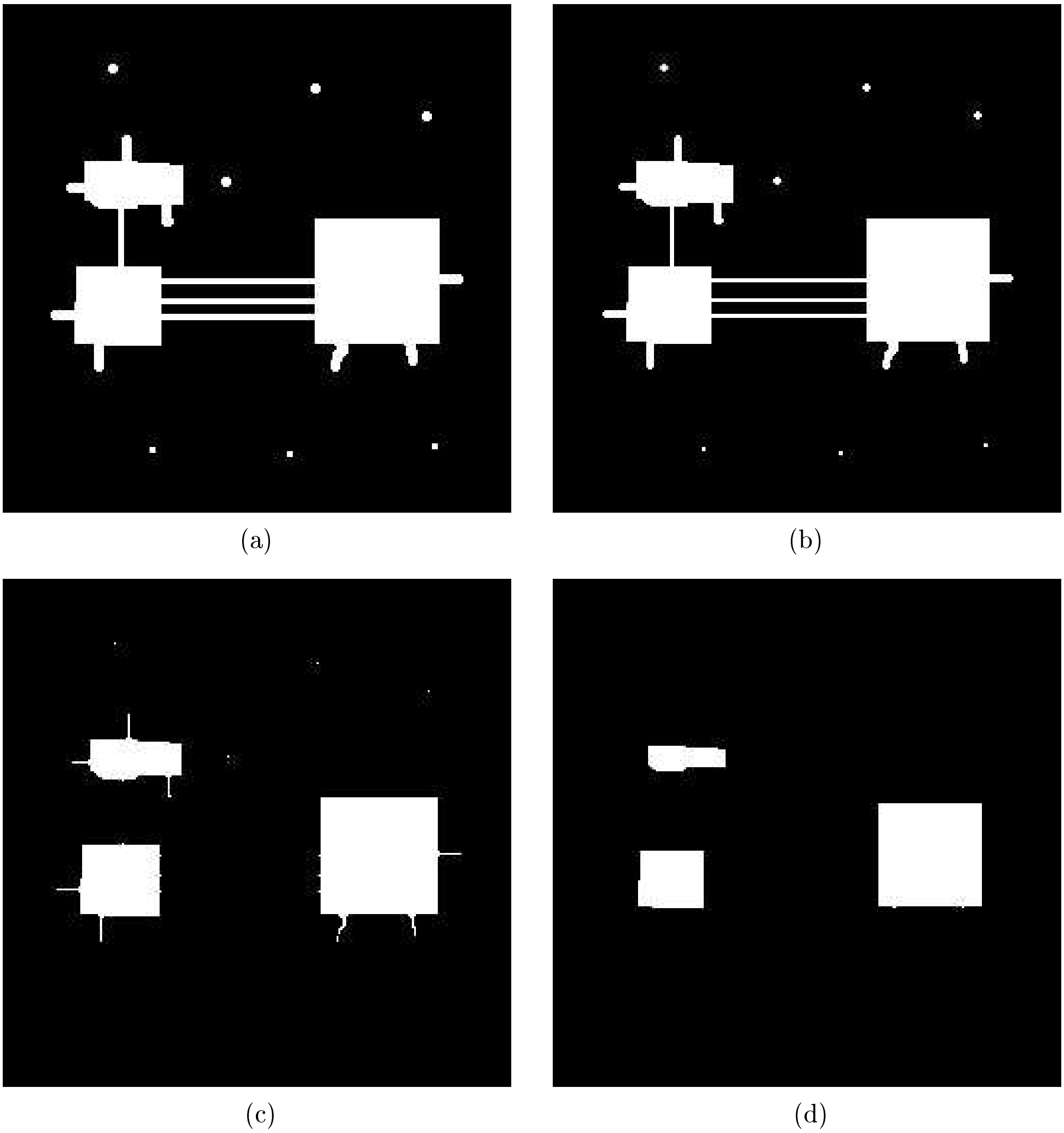


Figura C.3: Efecto de la Erosión sobre una imagen binaria: (a)Imagen original. (b)Erosión con EE circular de tamaño 3. (c)Erosión con EE circular de tamaño 5. (d)Erosión con EE circular de tamaño 7.

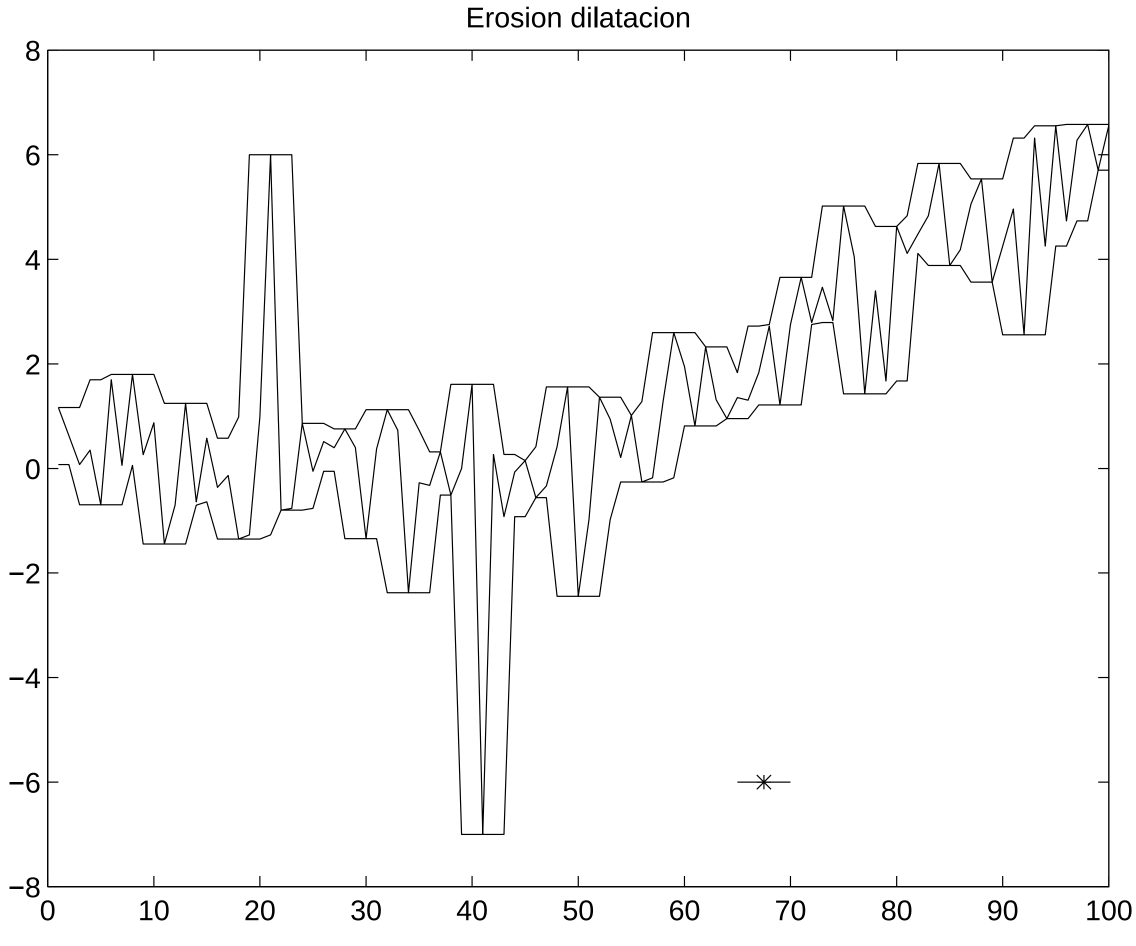


Figura C.4: Dilatación y erosión sobre una función. El elemento estructurante también esta representado.

C.2.2 Apertura y Cierre

La combinación mediante la aplicación concatenada de dos erosiones sucesivas o dos dilataciones sucesivas no aporta ningún operador conceptualmente nuevo, pero sí se obtienen operadoeres interesantes si se combinan mediante la aplicación concatenada erosión con dilatación o viceversa, llamadas respectivamente **apertura** y **cierre**.

Apertura La apertura de un conjunto X por un elemento estructurante B, denotada por $\gamma_B(X)$, se define como:

$$\gamma_B(X) = \delta_B(\epsilon_{\check{B}}(X))$$

lo que significa que primero se realiza una erosión y al resultado de la misma se realiza la dilatación. Si el elemento estructurante es simétrico, entonces la transposición del elemento estructurante no produce ningún cambio y se puede poner que $\gamma_B(X) = \delta_B(\epsilon_B(X))$.

La apertura cumple, entre otras, las siguientes dos propiedades:

- Creciente:

$$X \leq Y \Longleftrightarrow \gamma(X) \leq \gamma(Y)$$

- Anti-extensiva:

$$\gamma(X) \leq X$$

Cierre De igual manera, el cierre de un conjunto X por un elemento estructurante B, denotada por $\varphi_B(X)$, se define como:

$$\varphi_B(X) = \epsilon_B(\delta_{\check{B}}(X))$$

lo que significa que primero se realiza una dilatación y sobre el resultado de la misma se realiza la erosión. Si el elemento estructurante es simétrico, entonces la transposición del mismo no produce ningún cambio y se puede poner que $\varphi_B(X) = \epsilon_B(\delta_B(X))$.

El cierre cumple, entre otras, las siguientes dos propiedades:

- Creciente:

$$X \leq Y \Longleftrightarrow \varphi(X) \leq \varphi(Y)$$

- Extensiva:

$$\varphi(X) \geq X$$

Como consecuencia, se puede establecer el siguiente ordenamiento parcial:

$$\gamma(X) \leq X \leq \varphi(X)$$

Ambas definiciones se extienden a las funciones. Por brevedad, en ocasiones se simplificará la notación de la concatenación de funciones, de manera que una sucesión de transformaciones tal como $\mathcal{T}_B^1(\mathcal{T}_B^2(\ldots \mathcal{T}_B^n(X)))$ se escribirá como $\mathcal{T}_B^1\mathcal{T}_B^2 \ldots \mathcal{T}_B^n(X)$, e incluso se eliminará de la notación el elemento estructurante y/o la entrada X.

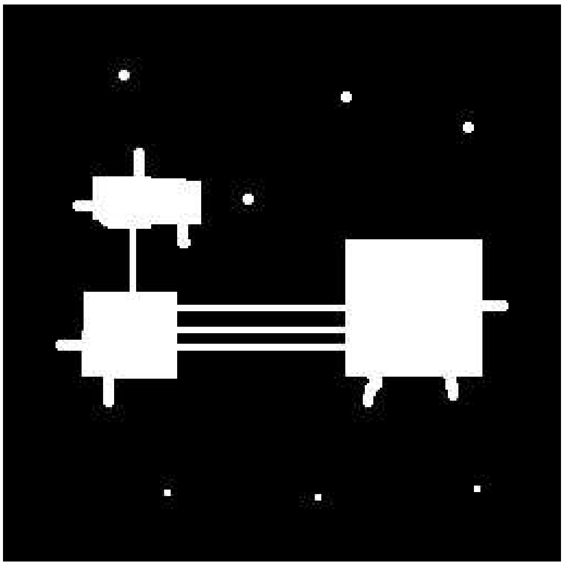
En el caso binario, las dos operaciones tienden a suavizar los contornos de los objetos. La apertura elimina las convexidades, mientras que el cierre elemina las concavidades. Asimismo si un objeto tiene dos partes unidas por un istmo de tamaño menor que el elemento estructurante, la apertura separará dicho objeto en dos desconectándolos, mientras que si dos objetos están separados por un espacio de fondo menor que el elemento estructurante, el cierre une ambos objetos para formar uno único conectándolos. En el caso de las funciones o imágenes de niveles de gris, como la apertura elimina las convexidades, se traduce en la eliminación de los picos (positivos), y como el cierre elimina las concavidades, se traduce en la eliminación de los valles.

De la misma manera que la pareja erosión-dilatación son *duales* también los son la pareja apertura-cierre, de forma que:

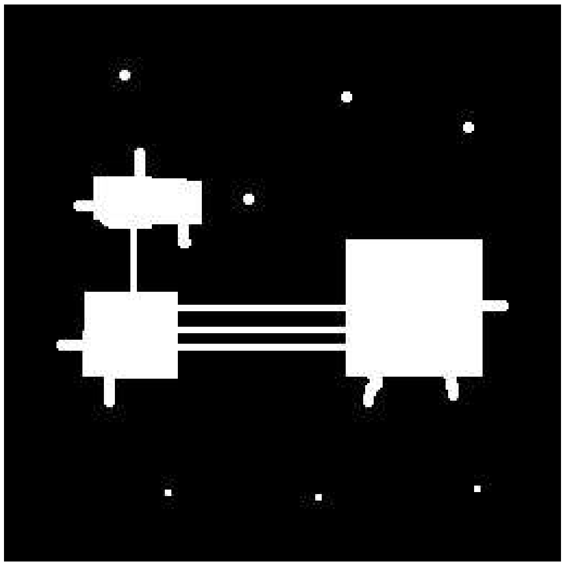
$$\gamma_B(X)^c = \varphi_B(X^c)$$

y a diferencia de la erosión y la dilatación, la apertura y el cierre son *idempotentes*, es decir:

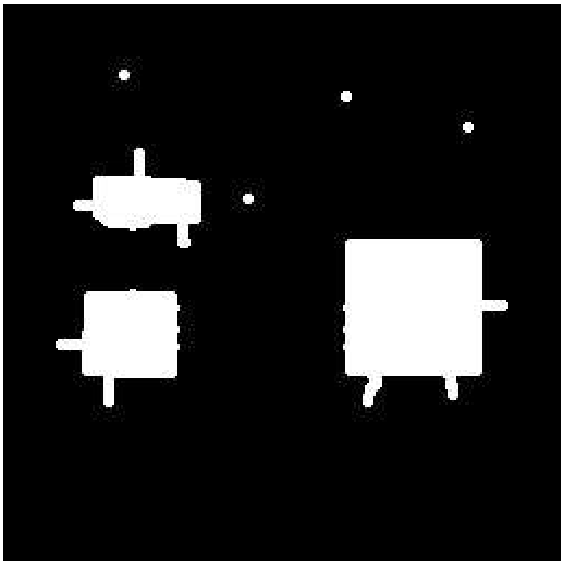
$$\gamma\gamma = \gamma \quad \varphi\varphi = \varphi.$$



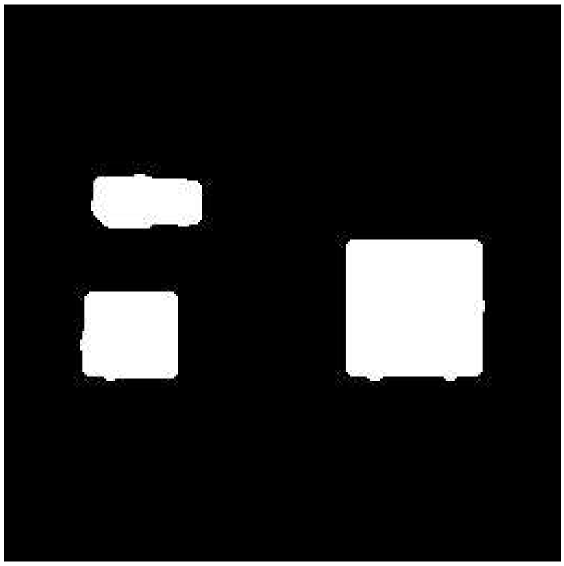
(a)



(b)



(c)



(d)

Figura C.5: Efecto de la Apertura sobre una imagen binaria: (a)Imagen original. (b)Apertura con EE circular de tamaño 3. (c)Apertura con EE circular de tamaño 5. (d)Apertura con EE circular de tamaño 7.

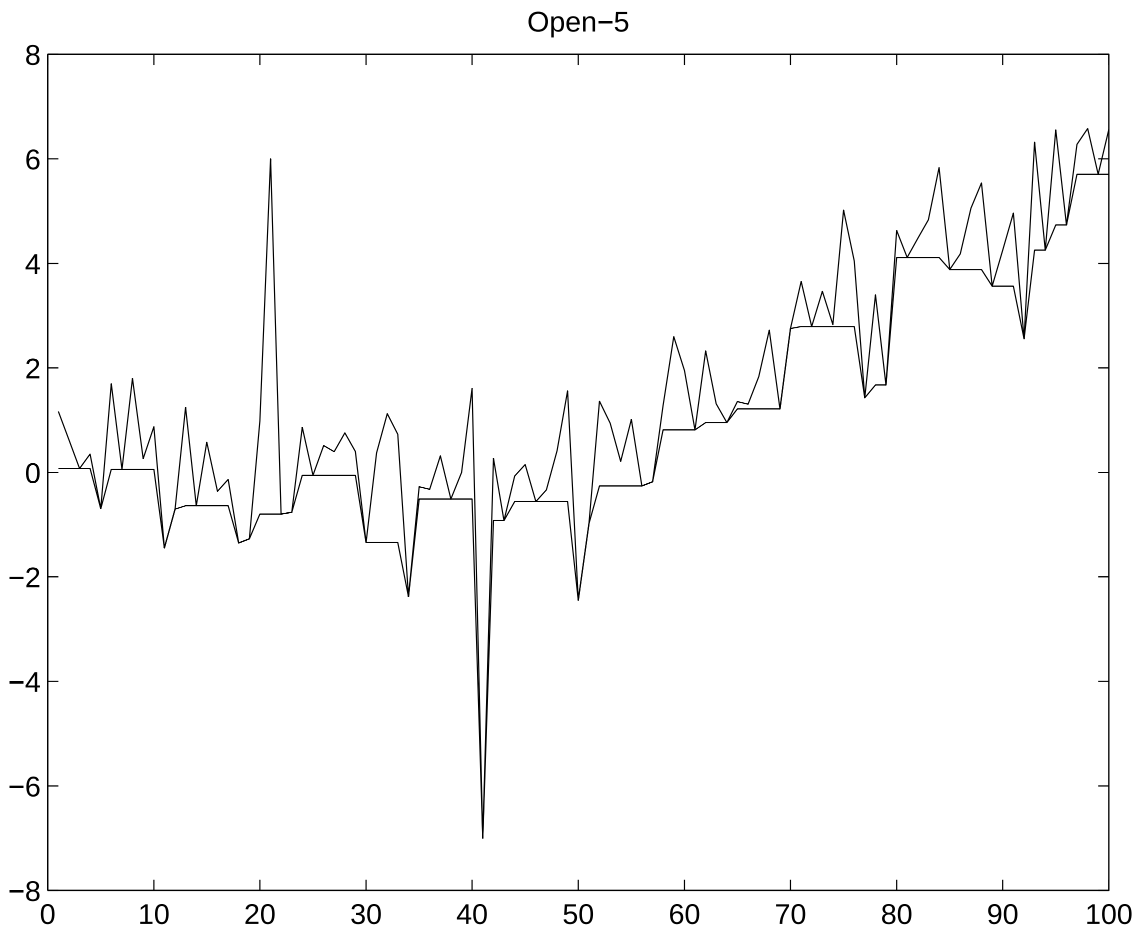


Figura C.6: Efecto de la apertura sobre una función.

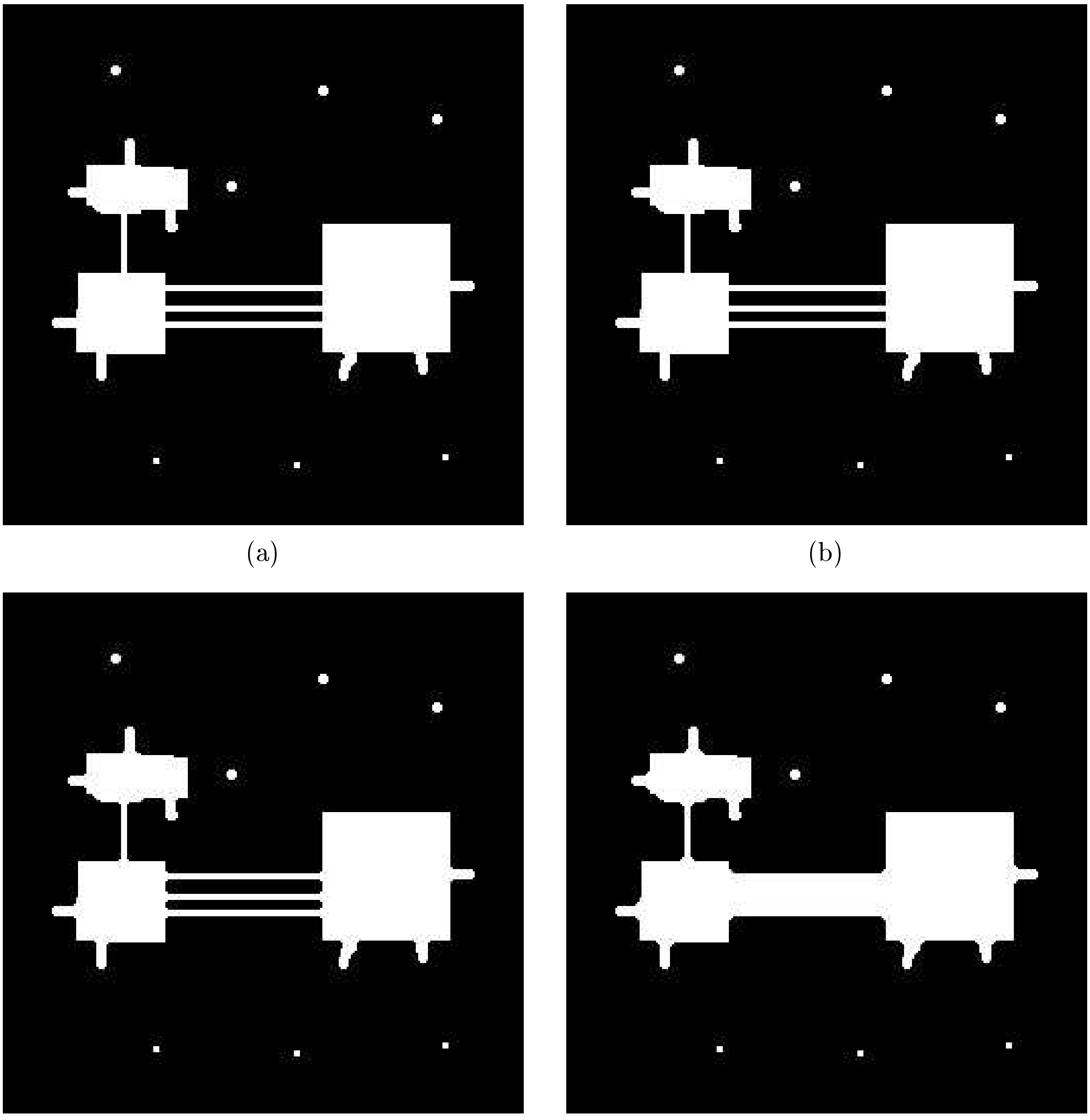


Figura C.7: Efecto del Cierre sobre una imagen binaria: (a)Imagen original. (b)Cierre con EE circular de tamaño 3. (c)Cierre con EE circular de tamaño 5. (d)Cierre con EE circular de tamaño 7.

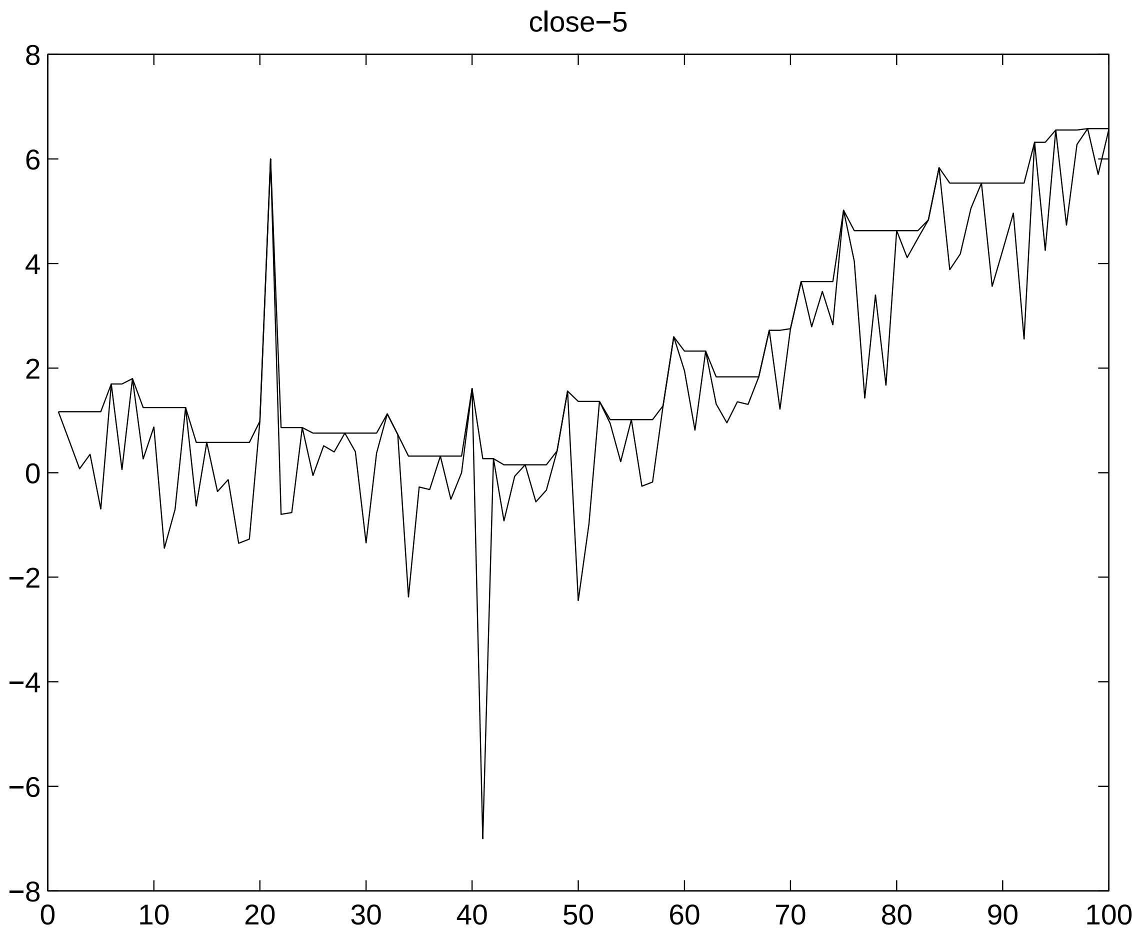


Figura C.8: Efecto del Cierre sobre una función.

C.2.3 Filtros morfológicos

El concepto de filtro está muy extendido en tratamiento de señal clásico y básicamente se refiere a cualquier proceso que un sistema realiza sobre una o varias entradas para generar una o varias salidas. En los sistemas lineales el filtrado va asociado a la convolución de la entrada con una determinada función (continua o discreta) que caracteriza el sistema. En morfología matemática el término *filtro* muy preciso, sujeto a la siguiente definición:

Filtro Un filtro morfológico es una transformación $\mathcal{T}$ creciente e idempotente.

$$\forall f, g \in \mathcal{F}, \quad f < g \Longrightarrow \mathcal{T}(f) < \mathcal{T}(g)$$

$$forall f \in \mathcal{F}, \quad \mathcal{T}(\mathcal{T}(f)) = \mathcal{T}(f)$$

La propiedad creciente es la más importante porque permite mantener la ordenación de los conjuntos o funciones tras la operación de filtrado.

Las operaciones básicas de dilatación y erosión no son filtros morfológicos proque no son idempotentes, salvo en el caso particular de que el elemento estructurante sea un único punto o elemento, situación en que las transformaciones se reducen a la identidad. Sin embargo la combinación de ambas, la apertura y el cierre definidos antes, sí son filtros morfológicos porque cumplen ser crecientes e idempotentes. En general las combinaciones de aperturas y cierres dan lugar a filtros morfológicos, y da lugar a una familia de transformaciones interesantes según la combinación se realice concatenando las transformaciones (filtros alternados secuenciales) o realizando todas las transformaciones en paralelo sobre una misma función de entrada y combinando las salidas mediante el supremo y el ínfimo (por ejemplo, el contraste morfológico, o el centro morfológico).

C.2.4 Residuos

La teoría de filtros morfológicos resalta las propiedades de creciente y de idempotencia, pero para nuestro proyecto han sido de mucha utilidad otra familia de transformaciones que estudia la diferencia entre dos o más transformaciones básicas. El concepto común en esta familia de transformaciones es el aspecto de *diferencia* también llamado *residuo*.

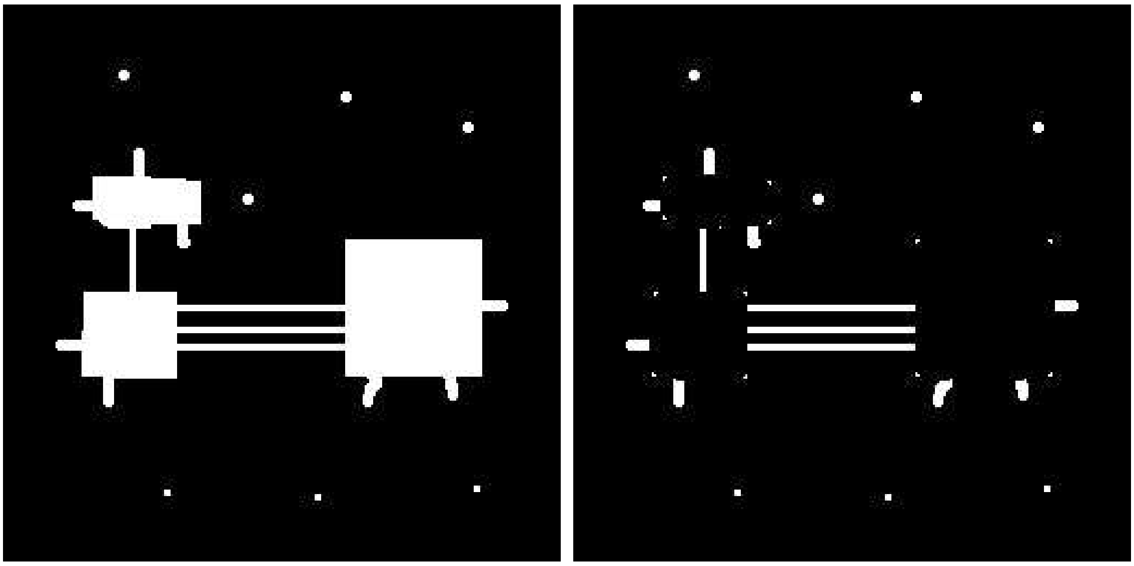
Entre ellas destacan las transformaciones **Top-Hat** que en su versión *blanca* o *positiva* se define particularizando la transformación $\mathcal{T}_1$ a la identidad y la transformación $\mathcal{T}_2$ a la apertura con un determinado elemento estructurante B:

$$TH_B^+(f) = f - \gamma_B(f)$$

como la apertura es anti-extensiva el Top-Hat positivo es siempre igual o mayor que cero, quedando extraídos los detalles positivos de la función f que han desaparecido al realizar la apertura. Si se quieren extraer los detalles negativos se puede realizar la versión *negra* o *negativa* del Top-Hat definida como:

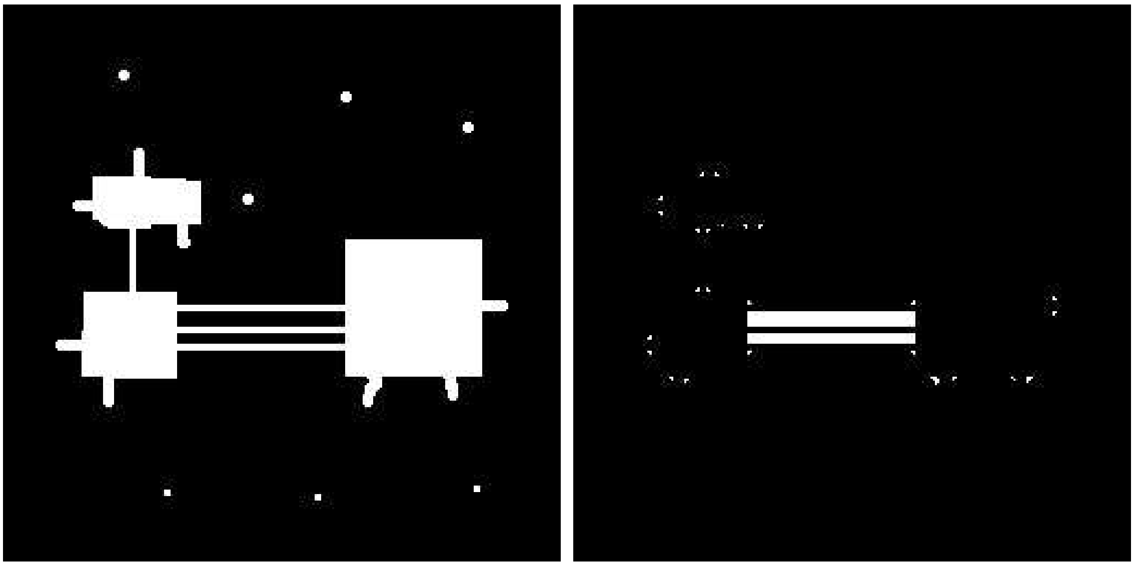
$$TH_B^-(f) = \varphi_B(f) - f$$

En el presente trabajo hemos estado interesados en buscar picos positivos, por eso hemos aplicado Top-Hat en su versión positiva.



(a) (b)

Figura C.9: Efecto del *Top-Hat* positivo sobre una imagen binaria: (a)Imagen original. (b) *Top-Hat* positivo con EE circular de tamaño 3.



(a) (b)

Figura C.10: Efecto del *Top-Hat* negativo sobre una imagen binaria: (a)Imagen original. (b) *Top-Hat* negativo con EE circular de tamaño 3.

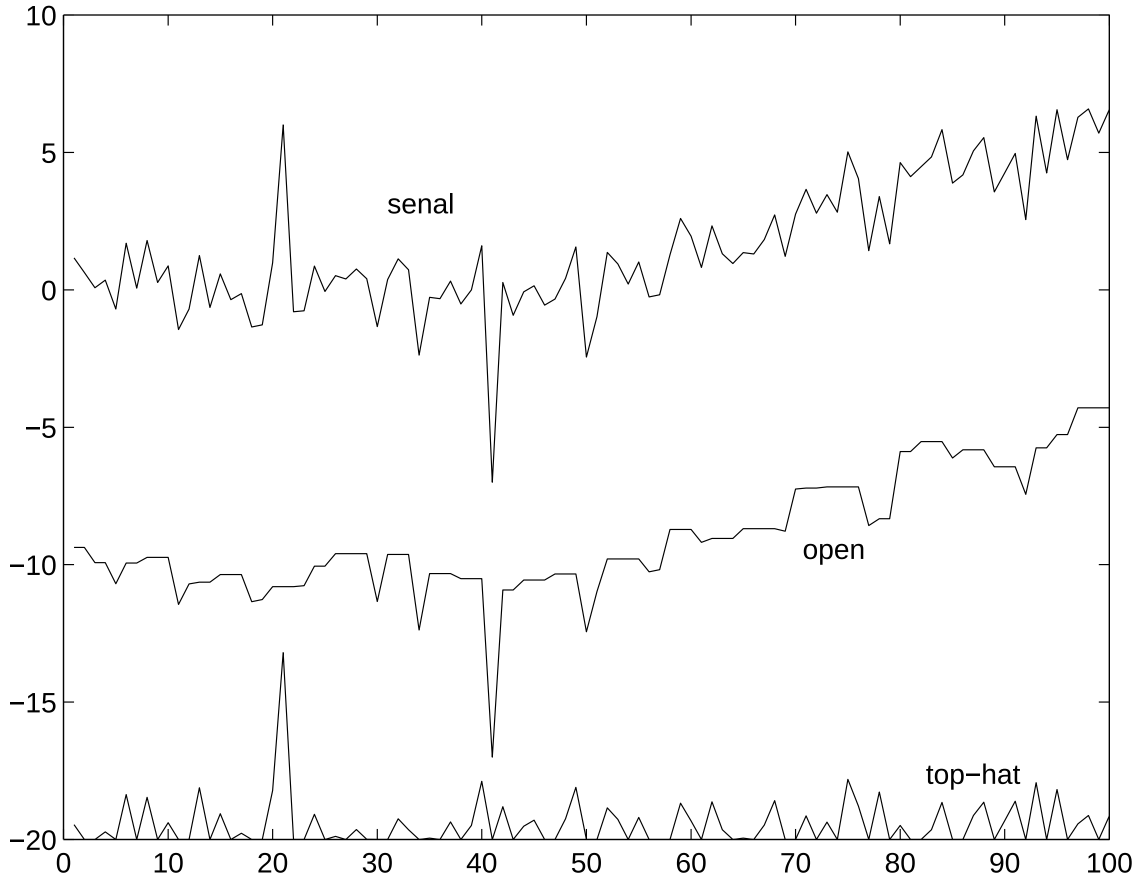


Figura C.11: Efecto del *Top-Hat* positivo sobre una función.

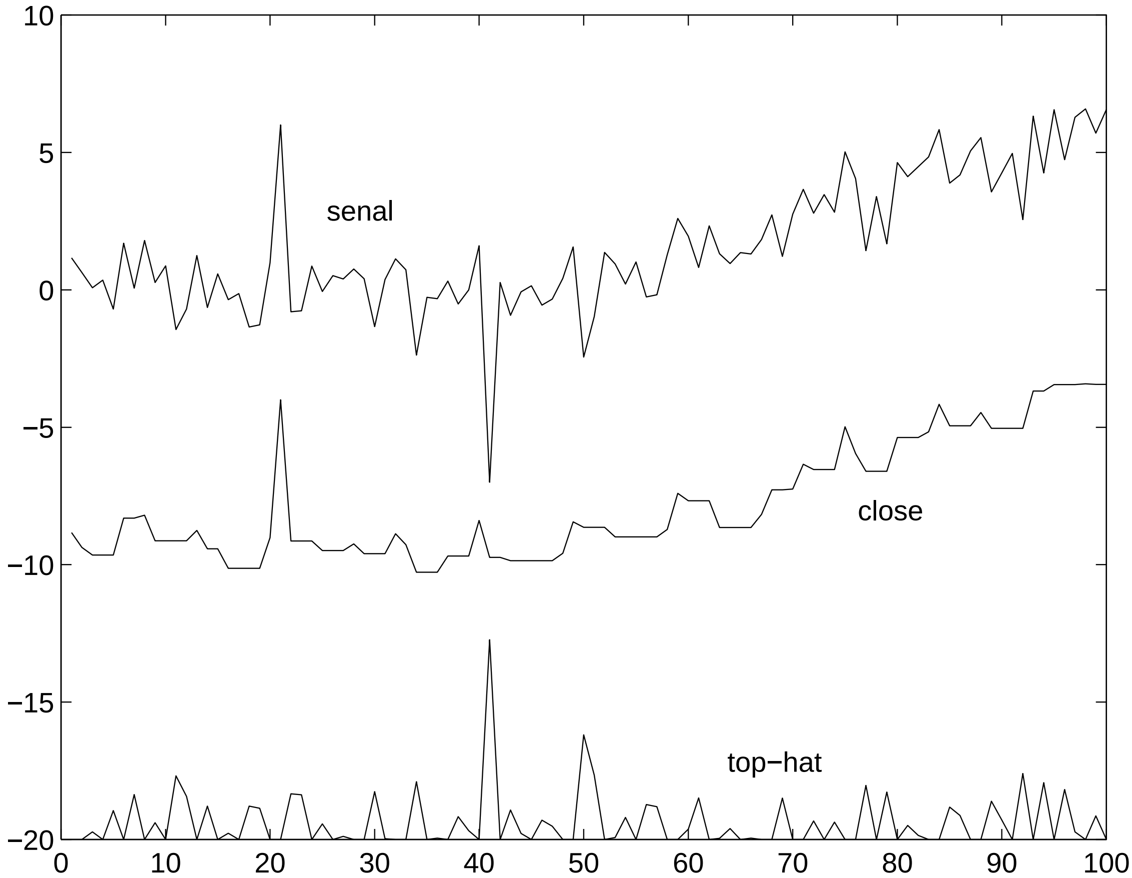


Figura C.12: Efecto del *Top-Hat* negativo sobre una función.

C.3 RECONSTRUCCIÓN GEODÉSICA

En las aplicaciones de tratamiento de imagen asumiendo un espacio euclideo la distancia entre dos puntos es la longitud del camino rectilíneo que los une. Hay aplicaciones donde el espacio que separa los dos puntos contiene partes que por algún motivo no pueden ser atravesadas pro el camino que une ambos puntos y consecuentemente la distancia ya nos es la distancia euclídeo. Este concepto lleva a definir una nueva distancia llamada *distancia geodésica.* Cuando tratamos con geodesia tenemos dos conjuntos, uno es el que contiene los dos puntos de los que queremos calcular la distancia geodésica y el otro es el que nos marca qué partes del espacio pueden ser atravesadas pro el camino que unirá los dos puntos. El primero se llama *marcador* y el segundo se llama *máscara geodésica.*

Distancia geodésica En el caso de los conjuntos (morfología binaria) las distancia geodésica $d_M(x, y)$ entre dos puntos $x, y \in M$ donde M (conjunto de $\mathcal{R}^2$) es la máscara geodésica, es la menor longitud de los caminos posibles que unen los puntos x e y dentro de M.

Si M está compuesto por dos o más zonas separadas entre sí (componentes no conexas) y los puntos x e y no pertenecen a la misma componente, no habrá camino posible dentro de M que los una. En este caso, por convención, la distancia geodésica ser infinita.

C.3.1 Dilatación y erosión geodésicas

En base a esto se puede definir la dilatación geodésica:

Dilatación geodésica binaria La dilatación geodésica binaria de tamaño n, denotada $\delta^n(M, X)$, de un conjunto X incluido en la máscara M se define como:

$$\delta^1(M, X) = \delta_{B1}(X) \cap M$$

$$\delta^n(M, X) = \underbrace{\delta_{B1}(\delta_{B1}(\ldots \delta_{B1}(X) \cap M) \ldots \cap M) \cap M)}_{n \;\; veces}$$

donde $\delta_{B1}(X)$ es la dilatación morfológica con un elemento estructurante formado por un disco de radio 1 según el tipo de conectividad.

Nótese que la dilatación geodésica de tamaño n no es la intersección de la dilatación morfológica de tamaño n con M.

$$\delta^n(M, X) \neq \delta^n(X) \cap M$$

La erosión geodésica es la transformación dual:

$$\epsilon^n(M,X) = M - \delta^n(M, M - X)$$

donde para conjuntos $M - X = M \cap X^c$.

Para las funciones o imágenes de niveles de gris la dilatación geodésica se define de una manera similar, donde ahora la entrada a dilatar es una función y las máscara geodésica (también llamada referencia) es una función mayor.

Dilatación geodésica binaria La dilatación geodésica binaria de tamaño n, denotada $\delta^n(M,X)$, de un conjunto X incluido en la máscara M se define como:

$$\delta^1(f_M, f) = inf(\delta_{B1}(f), f_M)$$

$$\delta^n(f_M, f) = \underbrace{\delta^1(f_M, \delta_1(f_M, \ldots \delta_1(f_M, f)\ldots))}_{n \ \ veces}$$

La erosión geodésica de funciones se obtiene por dualidad como:

$$\epsilon^n(f_M, f) = -\delta^n(-f_M, -f)$$

C.3.2 Reconstrucción geodésica

Una aplicación importante de la dilatación geodésica es implementar la operación llamada *reconstrucción*. En muchas ocasiones a la función a dilatar se le llama *marcador*. Si la función marcador es menor que la máscara, por definición el resultado de la dilatación geodésica siempre está dentro de alguna componente conexa de la máscara. Si la entrada a dilatar geodésicamente (conjunto o función) es distinto de cero en alguna o algunas componentes conexas de la máscara o referencia, la dilatación geodésica infinita dará lugar a la reconstrucción exacta de dicha o dichas componentes:

$$\forall X, Y \in \mathcal{P}(\mathcal{R}^2), \quad X \subset Y \Longrightarrow \delta^\infty(X,Y) = Y$$

En la práctica en vez de infinitas iteraciones, se dilata tantas veces como sea necesario hasta llegar a la idempotencia.

En el caso de funciones, la reconstrucción permite igualmente recuperar aquellas regiones de la función referencia marcadas por la imagen marcador. De nuevo la dilatación geodésica infinita en la práctica se reduce al número de iteraciones suficiente hasta llegar a la idempotencia. Las regiones son reconstruidas de manera que quedan incluidos en cada una de las regiones marcadas todos aquellos pixels que tienen un nivel inferior al máximo de la imagen marcador en cada región marcada. Si la imagen marcador contiene todos los máximos de la imagen referencia.

A partir de la reconstrucción se puede construir una familia de transformaciones morfológicas llamadas *transformaciones por reconstrucción* definidas como la composición de una transformación morfológica elemental y de la reconstrucción geodésica, por dilatación si la transformación es anti-extensiva, o por erosión si es extensiva.

Transformaciones con reconstrucción Sea $\mathcal{T}$ una transformación morfológica. Se define la correspondiente transformación con reconstrucción asociada a $\mathcal{T}$, $\mathcal{T}^{rec}$, de la siguiente manera:

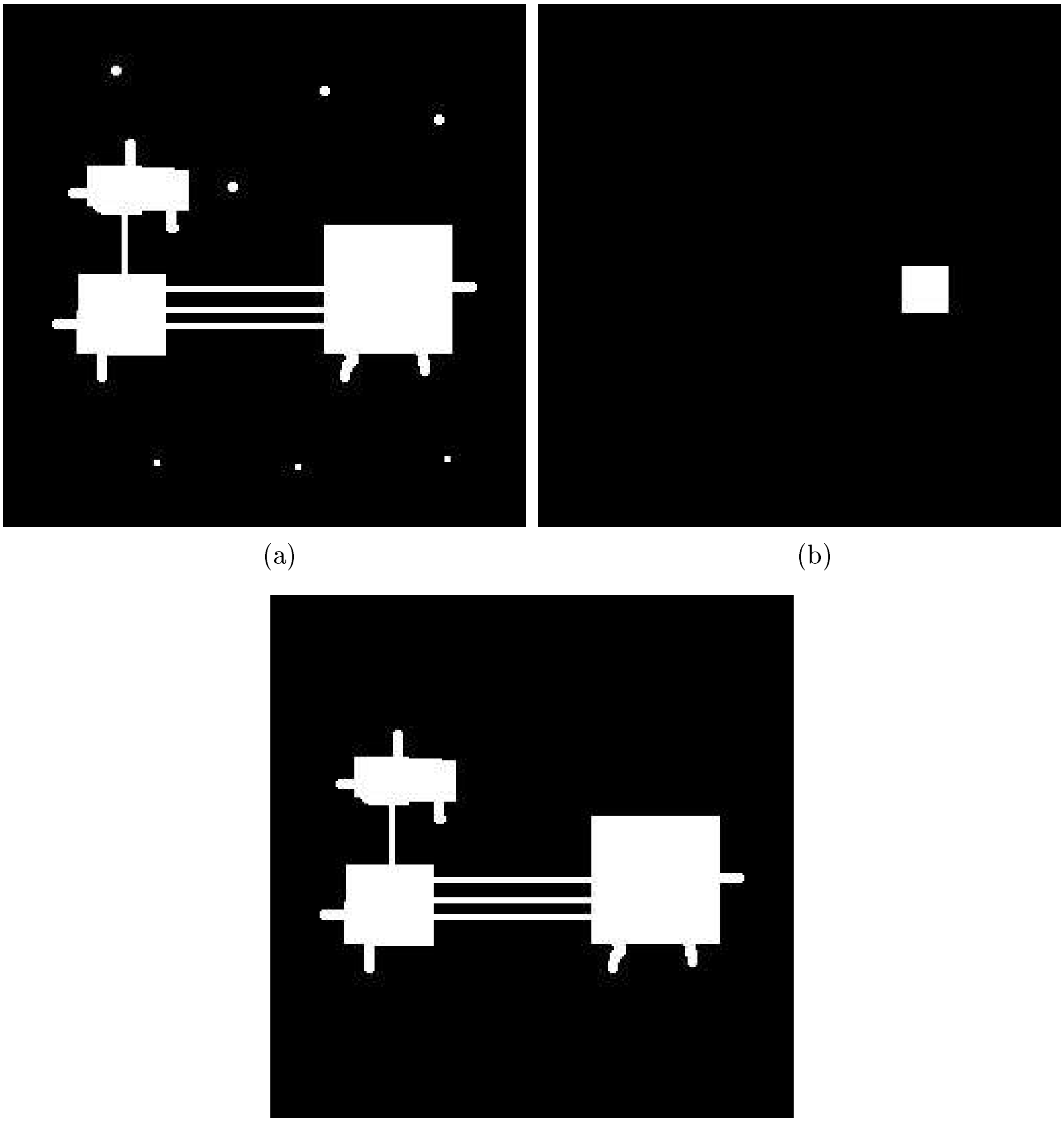
Si $\mathcal{T}$ es anti-extensiva

$$\mathcal{T}^{rec}(f) = \delta^{\infty}(f, \mathcal{T}(f))$$

Si $\mathcal{T}$ es extensiva

$$\mathcal{T}^{rec}(f) = \epsilon^{\infty}(f, \mathcal{T}(f))$$

Este tipo de transformaciones permite filtrar las imágenes, por ejemplo eliminando objetos menores que un cierto tamaño al tiempo que, gracias a la reconstrucción, no se modifican los contornos de los objetos.



(a) (b)

(c)

Figura C.13: Efecto de la reconstrucción geodésica sobre una imagen binaria: (a)Imagen referencia. (b)Imagen marcador (resultado de la dilatación con un elemento estructurante de tamaño 30, de la imagen referencia). (c)Imagen reconstruida.

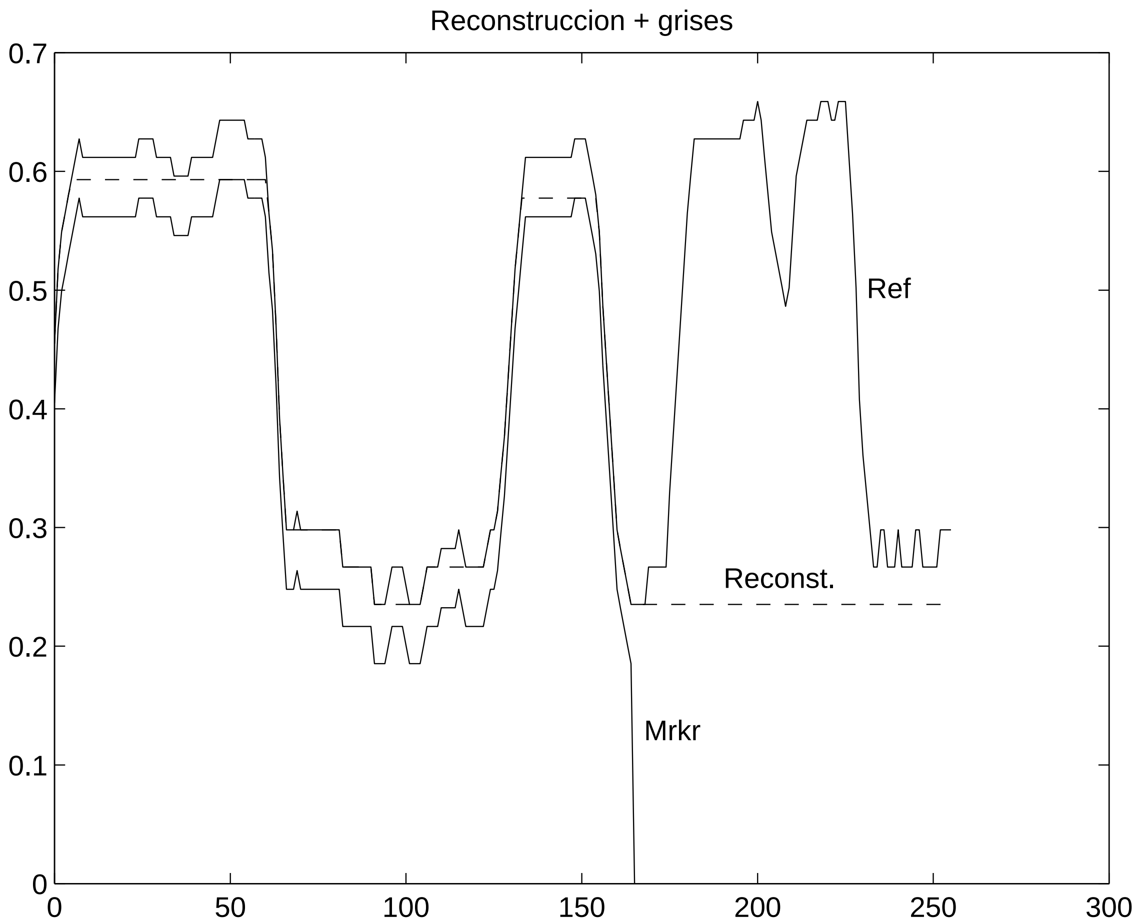


Figura C.14: Efecto de la reconstrucción geodésica sobre una función. En la figura aparecen la señal **referencia**, la señal **marcador** y el resultado de la **reconstrucción**.

C.4 REFERENCIA DE FUNCIONES DE MORFOLOGÍA 1-D PROGRAMADAS PARA PDIWin32

C.4.1 Introducción

El programa PDIWIN32 ya dispone de todas las funciones necesarias para realizar la mayoría de las operaciones de MM utilizadas para imágenes binarias e imágenes de grises.

Como parte del procesamiento realizado sobre los datos de nuestro proyecto, ha sido necesario incorporar una serie de funciones programadas en C, que implementasen los principales operadores morfológicos 1-D (aplicables a funciones).

A continuación, presentaremos cada una de estas funciones con la estructura siguiente:

- **Sinopsis:** Breve descripción de la función
- **Formato:** Sintaxis a utilizar en la llamada a la función y breve explicación de cada uno de los parámetros que intervienen.

C.4.2 Funciones de Morfología Matemática 1-D

Erode1D

- ☐ **Sinopsis:** Erosión de una señal unidimensional.
- ☐ **Formato:** `int Erode1D(float *xin,int tama_xin,int tamagno,float *yout);`
  - `float *xin`: Señal (array) a erosionar.
  - `int tama_xin`: Dimensión de xin.
  - `int tamagno`: Tamaño del elemento estructurante.
  - `float *yout`: Salida erosionada.

Dilate1D

- ☐ **Sinopsis:** Dilatación de una señal unidimensional.
- ☐ **Formato:** `int Dilate1D(float *xin,int tama_xin,int tamagno,float *yout);`
  - `float *xin`: Señal (array) a erosionar.
  - `int tama_xin`: Dimensión de xin.
  - `int tamagno`: Tamaño del elemento estructurante.
  - `float *yout`: Salida erosionada.

Open1D

- ☐ **Sinopsis:** Apertura de una señal unidimensional.
- ☐ **Formato:** `int Open1D(float *xin,int tama_xin,int tamagno,float *yout);`
  - `float *xin`: Señal (array) de entrada para la apertura.
  - `int tama_xin`: Dimensión de xin.
  - `int tamagno`: Tamaño del elemento estructurante.
  - `float *yout`: Salida de la apertura.

Close1D

- ☐ **Sinopsis:** Cierre de una señal unidimensional.
- ☐ **Formato:** `int Close1D(float *xin,int tama_xin,int tamagno,float *yout);`
  - `float *xin`: Señal (array) de entrada para el cierre.
  - `int tama_xin`: Dimensión de xin.
  - `int tamagno`: Tamaño del elemento estructurante.
  - `float *yout`: Salida del cierre.

OpenClose1D

- ☐ **Sinopsis:** Apertura y cierre concatenados de una señal unidimensional.
- ☐ **Formato:** `int OpenClose1D(float *xin,int tama_xin,int tamagno,float *yout);`
  - `float *xin`: Señal (array) a procesar.
  - `int tama_xin`: Dimensión de xin.
  - `int tamagno`: Tamaño del elemento estructurante.
  - `float *yout`: Salida procesada.

CloseOpen1D

- ☐ **Sinopsis:** Cierre y apertura concatenados de una señal unidimensional.
- ☐ **Formato:** `int CloseOpen1D(float *xin,int tama_xin,int tamagno,float *yout);`
  - `float *xin`: Señal (array) a procesar.
  - `int tama_xin`: Dimensión de xin.
  - `int tamagno`: Tamaño del elemento estructurante.
  - `float *yout`: Salida procesada.

FiltroAltSeq1D

- ☐ **Sinopsis:** Filtrado Alternado Secuencial de una señal unidimensional.

☐ **Formato:** `int FiltroAltSeq1D(float *xin,int tama_xin,int tamagno,int paso,bool close1st,float *yout);`

- `float *xin`: Señal (array) a realizar filtrado AltSeq.
- `int tama_xin`: Dimensión de xin.
- `int tamagno`: Tamaño del elemento estructurante mínimo.
- `int paso`: Paso de filtrado secuencial.
- `bool close1st`: A "Verdadero" indica que primero se hace un cierre.
- `float *yout`: Salida después del filtrado AltSeq.

TopHat1D

☐ **Sinopsis:** *Top-Hat* positivo de una señal unidimensional.

☐ **Formato:** `int TopHat1D(float *xin,int tama_xin,int tamagno,float *yout);`

- `float *xin`: Señal (array) a la que realizar el *Top-Hat*.
- `int tama_xin`: Dimensión de xin.
- `int tamagno`: Tamaño del elemento estructurante.
- `float *yout`: Salida procesada.

TopHatNeg1D

☐ **Sinopsis:** *Top-Hat* negativo de una señal unidimensional.

☐ **Formato:** `int TopHatNeg1D(float *xin,int tama_xin,int tamagno,float *yout);`

- `float *xin`: Señal (array) a la que realizar el *Top-Hat*.
- `int tama_xin`: Dimensión de xin.
- `int tamagno`: Tamaño del elemento estructurante.
- `float *yout`: Salida procesada.

ReconGeode1D

☐ **Sinopsis:** Reconstrucción geodésica de una señal unidimensional.

☐ **Formato:** `int ReconGeode1D(float *mrk,float *ref,int tama_xin,float *yout);`

- `float *mrk`: Señal (array) marcador de entrada.
- `float *ref`: Señal (array) referencia de entrada.
- `int tama_xin`: Dimensión de mrk y ref (deben tener la misma).
- `float *yout`: Salida resultado de la reconstrucción.